

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Estudo de dados industriais no contexto de Mobility SAP HANA

Mauro Monte Lira Rodrigues da Costa



Mestrado Integrado em Engenharia Informática e Computação

Orientador: Professor João José Pinto Ferreira

5 de Setembro de 2017

Estudo de dados industriais no contexto de Mobility SAP HANA

Mauro Monte Lira Rodrigues da Costa

Mestrado Integrado em Engenharia Informática e Computação

Resumo

A computação em memória tem ganho uma importância ímpar nas últimas décadas com a emergência de diversos sistemas gestores de bases de dados com uma capacidade de processamento e eficiência difíceis de igualar, com respostas em tempo real para volumes de dados gigantescos e complexos. Estes dados consomem bastantes recursos computacionais, mas, na verdade, grande parte deles são descartados e ignorados pelas organizações, levando a um pobre aproveitamento da informação.

Ao longo dos últimos anos, as arquiteturas dos sistemas de informação têm-se revelado mais e mais complexas. A quantidade de informação necessária à gestão de uma organização tende a crescer de uma forma muito célere, originando tempos de resposta a consultas da base de dados muito indesejáveis, inoportunos e custosos em tempo e, conseqüentemente, em dinheiro. O surgimento desta tecnologia inovadora foi possível devido aos avanços cruciais desenvolvidos ao longo das últimas décadas, ao nível do hardware, nomeadamente um aumento da capacidade de memória RAM e um maior poder de processamento da CPU, com a emergência de processadores com vários núcleos que maximizam o uso de paralelismo, indispensável em sistemas como o SAP HANA.

O foco desta dissertação será o estudo de dados industriais relativos a uma indústria de construção, onde os tempos de resposta estão a revelar-se muito longos e penosos. De forma a avaliar o impacto no negócio de uma solução em memória, neste caso, o SAP HANA (que opera 100% em memória RAM) será comparado com a solução existente, o SAP R/3, através da medição de tempos de execução de determinadas *queries* analíticas. Espera-se, de uma base de dados deste tipo, obter tempos de resposta às consultas bastante mais reduzidos e, desta forma, criar vantagem competitiva para a organização. Pretende-se ainda mostrar a possível criação de valor para o negócio adjacente a estes sistemas, nomeadamente, ao nível da gestão de inventário e listagem de materiais. Para isso, irá ser desenvolvida uma aplicação *web*, com a função de maximizar o auxílio e a flexibilidade na toma de decisões no mundo empresarial.

Com o crescente impacto dos dispositivos móveis, a computação em memória será a chave para a sua integração com o negócio das empresas, permitindo aos administradores tomarem decisões em qualquer altura e em qualquer lugar, assegurando uma vantagem competitiva da empresa num futuro cada vez mais presente.

Abstract

Memory computing has gained unprecedented importance in recent decades with the emergence of many database management systems with a processing and efficiency capacity difficult to match, with real-time answers to a huge and complex volume of data. These data consume a lot of computational resources, but, actually, many of them are discarded and ignored by the organizations, what takes to a poor use of information.

Over the last few years, the architectures of information systems revealed more and more complex. The amount of information needed to manage an organization tends to grow very fast, resulting in response times to database queries which are very undesirable, inopportune and costly in time and, consequently, expensive. The emergence of this innovative technology was possible due to the crucial advances that have been developed over the last decades at the hardware level, namely an increase of the capacity of memory RAM and a greater power of processing of the CPU, with the emergence of multi-core processors that maximize the use of parallelism, crucial in systems such as SAP HANA.

The focus of this dissertation will be the study of industrial data related to a construction industry, where the response times are proving to be very long and painful. In order to evaluate the business impact of an in-memory solution, in this case, SAP HANA (which operates 100% in RAM) will be compared to the existing solution, SAP R/3, by measuring the execution times of certain analytic queries. It is expected from a database of this type to obtain response times to the queries considerably lower and, in this way, create competitive advantage for an organization. It is also intended to show the possible value creation for the adjacent business to these systems, namely at the level of inventory management. For this, a web application will be developed, in order to maximize the aid and flexibility of decision-making in the business world.

With the growing impact of mobile devices, memory computing will be the key to their integration with the company business, allowing administrators to make decisions anytime and anywhere, ensuring a competitive advantage of the company in an increasingly present future.

Agradecimentos

Desde 2011, toda esta jornada se revelou uma experiência muito gratificante. O sentimento no término do curso é de total realização e satisfação, tanto a nível pessoal como profissional, graças à Faculdade de Engenharia da Universidade do Porto. Um muito obrigado a esta grande instituição e votos na continuação da formação de alunos e seres humanos de excelência.

Uma palavra especial para toda a minha família, que sempre me apoiou incondicionalmente desde o primeiro dia. Os meus pais, *Eduarda* e *Rui* e a minha irmã *Patrícia* que tornaram tudo isto possível. Um agradecimento especial à *Tânia* por todo o apoio e ajuda no foco da realização deste trabalho.

A realização da investigação não seria de todo possível sem a ajuda da *AMT Consulting*, nomeadamente a equipa dos *AMT Labs*. O seu conhecimento e experiência foram preciosos e o seu bom ambiente facilitaram todo o trabalho. Endereço um agradecimento especial ao *Fábio Silva* e *Filipa Galiza*, assim como ao *Rogério Pereira*, representante da empresa cliente, pelo apoio prestado.

Em último lugar, mas não menos importante, um muito obrigado ao meu orientador FEUP, o Professor *João José Pinto Ferreira*, por todo o apoio e orientação ao longo da realização da dissertação.

Mauro Costa

*“I must not fear.
Fear is the mind-killer.
Fear is the little-death that brings total obliteration.”*

Frank Herbert, *Dune*

Conteúdo

1	Introdução	1
1.1	Contexto/Enquadramento	2
1.2	Motivação e Objetivos	3
1.3	Estrutura da Dissertação	3
2	Computação em memória principal	5
2.1	Background	6
2.1.1	Avanços no hardware	6
2.1.2	Codificação do dicionário	7
2.1.3	Técnicas de compressão	8
2.1.4	Tipos de <i>queries</i>	9
2.1.5	<i>Layout</i> dos dados em memória principal	9
2.1.6	Operações da base de dados	10
2.1.7	Paralelismo	12
2.1.8	Cache de agregação	13
2.1.9	<i>Differential Buffer</i>	14
2.2	Plataformas existentes	14
2.2.1	Microsoft SQL Server 2014 (Hekaton)	15
2.2.2	Oracle Database 12c	15
2.2.3	SAP HANA	16
2.2.4	SAP R/3	18
2.3	Vantagem competitiva e conclusões	19
3	Revisão da Literatura	21
3.1	Discussão crítica	23
3.2	Conclusões	24
4	Metodologia de Investigação	27
4.1	Pergunta de Investigação	27
4.2	Paradigma <i>Design Science - Information Systems Research Framework</i>	28
4.3	Enquadramento do ambiente de experimentação	29
4.3.1	Desafios de <i>Big Data</i>	30
4.3.2	Dados industriais (<i>shop floor data</i>) e criticidade de tempo	30
4.3.3	Perspetiva de solução	31
4.4	Experimentação	32
4.4.1	Dados	32
4.4.2	Origem e Tipo	32
4.4.3	Dimensão	33

CONTEÚDO

4.4.4	Validade	34
4.4.5	Método de Avaliação	34
5	Implementação e Ensaio Experimental	37
5.1	Implementação	37
5.1.1	Aplicação em SAP R/3	38
5.1.2	Aplicação em SAP HANA	39
5.2	Ensaio Experimental	40
5.2.1	Queries de teste	41
5.2.2	Servidores e erros esperados	42
6	Discussão dos resultados	45
6.1	SAP R/3	45
6.2	SAP HANA	46
6.3	Validade e discussão	47
6.4	Criação de vantagem competitiva	49
6.5	Limitações do estudo	50
7	Conclusões e Trabalho Futuro	53
7.1	Trabalho Futuro	54
	Referências	55
A	Computação em memória - conceitos adicionais	59
A.1	Particionamento	59
A.2	<i>Logging e Recovery</i>	60

Lista de Figuras

2.1	Visão conceitual da hierarquia de memórias, com o seu desempenho e capacidade [Pla]	6
2.2	Exemplo da criação das estruturas dicionário e vetor de atributos para a coluna <i>fname</i> [Pla]	7
2.3	Operações em linha e coluna para ambos os tipos de <i>layout</i> : orientado a linha, orientado a coluna e híbrido, da esquerda para a direita [Pla]	10
2.4	Tipos de paralelismo em bases de dados em memória principal [Pla]	13
2.5	Conceito de <i>differential buffer</i> [Pla]	14
2.6	Arquitetura da base de dados HANA [FCP ⁺ 12]	18
2.7	Criação de valor através duma tecnologia em memória DRAM [vBDRM13]	19
4.1	<i>Information Systems Research Framework</i> , adaptada de [HMPR04]	29
4.2	Diagrama UML representando a organização dos dados	33
5.1	Visão da aplicação de logística de materiais em SAP HANA	39
5.2	Diagrama de sequência que esquematiza o fluxo da aplicação	40
6.1	Tempo médio de execução das <i>queries</i> , para cada grupo de resultados	49

LISTA DE FIGURAS

Lista de Tabelas

2.1	Principais diferenças entre OLTP e OLAP [Che]	9
2.2	Layouts possíveis para uma abordagem em memória principal [Pla]	10
3.1	Revisão da Literatura	21
4.1	Número de registos das tabelas	34
5.1	Conjunto de testes de <i>queries</i> analíticas	41
6.1	Média dos tempos de execução em SAP R/3	45
6.2	Média dos tempos de execução em SAP HANA	46
6.3	Média dos tempos de execução em ambos os sistemas	48
6.4	Média dos tempos de execução (ms) para os três tipos de resultados	48

LISTA DE TABELAS

Abreviaturas

CPU	Central Processing Unit
CEO	Chief Executive Officer
ERP	Enterprise Resource Planning
TCO	Total Cost of Ownership
DRAM	Dynamic RAM
RAM	Random Access Memory
ACID	Atomicity, Consistency, Isolation, Durability
OLTP	On-line Transactional Processing
OLAP	On-line Analytical Processing
SQL	Structured Query Language
SGBD	Sistema Gestor de Bases de Dados
SGA	Shared Global Area
SSD	Solid State Drive
MDX	MultiDimensional eXpressions
MVCC	Multi-version Concurrency Control
ABAP	Advanced Business Application Programming
ISRF	Information Systems Research Framework
SCM	Supply Chain Management
MES	Manufacturing Execution System
SAPUI	SAP User Interface
HTML	HyperText Markup Language
IQR	Inter Quartile Range
AJAX	Asynchronous Javascript And XML
JSON	JavaScript Object Notation
XML	eXtensible Markup Language
XSJS	XS JavaScript
MVC	Model-View-Controller
IDE	Integrated Development Environment
VPN	Virtual Private Network
GB	Gigabyte
CIO	Chief Information Officer

Capítulo 1

Introdução

Vivemos numa era onde a utilização de dispositivos móveis, desde *smartphones* a *tablets*, faz cada vez mais parte do nosso quotidiano e a sua integração com o negócio e os seus processos está a tornar-se muito relevante e crítica para o futuro das empresas. A integração com os seus sistemas de informação e a sua forma de armazenamento dos dados, desde bases de dados relacionais a bases de dados em memória principal, é também um aspeto determinante neste caminho para o sucesso.

A informação que uma empresa necessita para a gestão do seu negócio tende a aumentar de forma exponencial em relativamente poucos anos, levando a grandes volumes de dados não aproveitados. Muitas das vezes, estes dados são simplesmente ignorados pelas organizações mas permanecem a consumir recursos computacionais, como capacidade de armazenamento e recursos da CPU. Esta quantidade gigantesca de informação é conhecida pelo termo *Big Data* e é considerado um dos maiores desafios, atuais e futuros, que tem despertado a atenção de investigadores e pessoas com cargos de tomada de decisões, tanto em governos como em empresas [PCZ14]. O aproveitamento desta informação pode trazer inúmeras vantagens às empresas, desde um aumento na eficiência operacional e um melhor serviço aos clientes até à identificação e desenvolvimento de novos produtos e serviços e uma maior rapidez na chegada ao mercado. [PCZ14]

Sempre que o tempo de resposta a uma determinada consulta é elevado (acima de 10 segundos) não se espera que um utilizador, sendo ele o CEO, um administrador ou um simples funcionário da empresa esteja disposto a aguardar todo esse tempo. Pelo contrário, isso potencia uma maior probabilidade de perda de foco na tarefa em causa, tornando-se difícil regressar à mesma, aquando da resposta do sistema [Nie]. Por outro lado, se os tempos de resposta forem em tempo real, um dos principais aspetos em consideração na implementação do SAP HANA, [FCP⁺12] os mesmos utilizadores são capazes de tomar decisões a qualquer hora e em qualquer lugar e, assim, aumentar drasticamente a flexibilidade no suporte na adoção de estratégias, desde decisões mais operacionais até assuntos mais estratégicos do ponto de vista empresarial. Sendo a velocidade um fator

chave para o sucesso do negócio, tempos de resposta para a execução de *queries* para tomada de decisões devem estar entre milissegundos e segundos. [BTMF12]

Ao longo das últimas décadas, de forma a analisar e a lidar com este crescendo do fluxo de informação, foram surgindo tecnologias e sistemas, tais como, bases de dados relacionais, armazéns de dados ou ERP's. Apesar da sua grande utilidade, estes têm revelado tempos de consulta demasiado custosos, para volumes gigantescos de dados, e uma solução em memória principal promete ser o caminho para o futuro sucesso de empresas críticas em tempo, onde a sua gestão envolve volumes enormes de informação.

A SAP lançou a plataforma HANA em 2010, uma plataforma de processamento de dados em memória principal, capaz de processar enormes volumes de dados com uma grande eficiência, apesar de até hoje ter sofrido variadas alterações [SFGL13]. Com este sistema esperam-se obter respostas em tempo real para *queries* analíticas complexas e *queries* transacionais (mais simples), assim como agregações *on-the-fly*. Apesar da melhoria ser substancialmente superior no que concerne ao processamento analítico [BDMB16], poupa-se uma quantidade abismal de tempo o que se reflete numa vantagem competitiva e na criação de valor para as empresas. Contudo, a transição de uma base de dados tradicional, centrada em disco, para uma base de dados centrada em memória, é considerada um investimento dispendioso [BDMB16]. O surgimento deste tipo de sistemas foi possível graças a avanços no *hardware*, ao nível da capacidade de memória e poder de processamento, com o emergir de processadores com vários núcleos. [Pla]

A dissertação foca-se em fazer um estudo de dados de uma empresa industrial portuguesa (*shop floor data*), comparando o desempenho obtido no sistema atualmente adotado pela organização com uma solução em memória, neste caso, a plataforma SAP HANA, otimizada, de forma a reduzir drasticamente os tempos de execução das *queries*. Pretende-se otimizar a consulta de relatórios SAP usados para gestão de inventário e descrição de movimentos de materiais, desenvolvendo uma aplicação que auxilie os administradores na tomada de decisões e agilize todo este processo. Naturalmente, negócios não críticos em tempo caem fora da esfera de uma abordagem desta natureza.

1.1 Contexto/Enquadramento

A dissertação insere-se na área científica de Sistemas de Informação, com ênfase na Gestão Empresarial. Será desenvolvida em parceria com a AMT Consulting, uma empresa de consultoria de tecnologias de informação que tem como principal foco a gestão de capital humano, com instalações em Lisboa, no Porto e em São Paulo. Este estudo pretende demonstrar as capacidades do sistema SAP HANA, no processamento de volumes gigantescos de informação, de forma a apresentar e reforçar o seu valor para o negócio, sobretudo, como solução para uma gestão melhorada do inventário, através da redução significativa dos tempos de consulta.

1.2 Motivação e Objetivos

Em qualquer negócio, o seu planeamento, gestão e estratégias adotadas, aliadas à tomada de decisões constituem fatores críticos no caminho para o sucesso. Esta tomada de decisões revela ter uma importância enorme na gestão de uma empresa, desde uma melhor utilização de recursos, uma ajuda na concretização dos objetivos até uma facilitação da inovação, refletindo-se numa vantagem competitiva para a organização [Akr]. Um dos objetivos da dissertação foca-se em auxiliar e facilitar esta tomada de decisões por parte das empresas, ao nível de gestão de inventário, tornando-a mais eficiente e flexível.

Espera-se que o SAP HANA tenha um grande impacto nesta gestão de decisões, uma vez que é a base para a próxima geração de plataformas de aplicações, permitindo implementar grandes aplicações de negócios, que permitam um desenvolvimento flexível e eficiente, por um lado, e implementações robustas e não disruptivas, assim como, operações com baixo TCO (*Total Cost of Ownership*), por outro lado. [SFL⁺12]

Mostrar o potencial do SAP HANA em termos de eficiência e capacidade de processamento, demonstrar o seu valor para o negócio e avaliar o seu impacto são os principais objetivos da dissertação. Uma otimização dos relatórios SAP produzidos pelo cliente irá ser feita, de forma a tornar as consultas à base de dados muito mais céleres, contrastando com os tempos demorados do programa em SAP R/3. Será desenvolvida uma aplicação *web* para gestão do inventário (*shop floor data*), como prova de conceito. Os dados a serem usados no estudo serão dados reais fornecidos pelo cliente, uma empresa industrial do setor de construção cuja identidade não deve ser revelada no âmbito desta dissertação.

1.3 Estrutura da Dissertação

Para além desta introdução, esta dissertação irá abordar o problema em causa, conceitos adjacentes à computação em memória, perspetiva de solução e correspondente metodologia, assim como detalhes de implementação e experimentação conduzidas. No capítulo 2 são abordados os principais conceitos da computação em memória e são apresentados sistemas de gestão de dados em memória principal. De seguida, no capítulo 3 é feita a revisão da literatura e é apresentada uma discussão crítica dos resultados. No capítulo 4 é descrita a metodologia de investigação utilizada e é formulada a pergunta de investigação. São também referidas informações sobre os dados usados no estudo e o enquadramento no ambiente experimental. Os detalhes da implementação e a experimentação conduzida serão apresentados no capítulo 5 e os seus resultados e principais limitações no capítulo 6. Por último, no capítulo 7, são apresentadas as conclusões e reflexões acerca do trabalho desenvolvido, incluindo aspetos potenciais de um trabalho futuro.

Introdução

Capítulo 2

Computação em memória principal

Nos últimos 30 anos, os sistemas gestores de bases de dados tradicionais, centrados em disco (memória persistente), foram considerados um componente fulcral nos sistemas e aplicações empresariais. Talvez se pensasse que esta solução pudesse ser de difícil substituição, mas, sobretudo, na última década e, muito devido aos avanços na tecnologia ao nível do *hardware*, emergiu uma nova abordagem que promete uma alteração no paradigma de armazenamento dos dados, com toda a gestão dos mesmos a ser feita em memória principal (DRAM). Uma separação dos sistemas transacionais e analíticos era pretendida de forma a aumentar o desempenho e permitir uma maior flexibilidade. [Pla]

As primeiras implementações desta inovadora tecnologia usaram um formato de armazenamento em linha, o que não revelou vantagens muito significativas em relação aos sistemas tradicionais de memória persistente. Só com um formato orientado a coluna é que foram despoletadas as potencialidades desta abordagem, permitindo a um sistema ser capaz de manipular capacidades analíticas e transacionais em conjunto, e assim, tornando desnecessário a replicação dos dados, processo este custoso em recursos computacionais. [Pla09]

Apesar dos princípios de *design* tidos em conta neste tipo de sistemas, de forma a alcançarem um desempenho ótimo, estes enfrentam alguns desafios em comparação com as bases de dados tradicionais. Um dos principais desafios está na garantia da durabilidade das transações, devido ao seu *design* baseado em RAM, enquanto que nas bases robustas tradicionais, devido ao seu *design* baseado em disco, a natureza ACID das transações é, naturalmente, garantida. Várias técnicas são aplicadas pelos sistemas para garantir esta durabilidade, como utilização de memória principal não volátil, *logging* de transações e replicação dos dados para outros nós disponíveis em tempo real. [GVV13]

Como principais capacidades, esta tecnologia recente possui cinco características principais que a potenciam: os dados são completamente guardados em memória, o que diminui, consideravelmente, os tempos de acesso; maximização da utilização de recursos computacionais através de paralelismo; suporta abordagens de armazenamento misto (coluna e em linha) para lidar com

workloads OLTP e OLAP; utilização de técnicas de compressão para reduzir o tamanho dos *datasets* e a adoção de uma abordagem inovadora, o *Insert-Only*, que será visto mais à frente, como forma de substituir operações de *update* e *delete*. [vBDRM13]

2.1 Background

2.1.1 Avanços no hardware

Nos inícios do ano 2000 foram introduzidas arquiteturas com vários núcleos, iniciando-se uma tendência, cada vez maior, de utilização de paralelismo. Atualmente, uma placa de computação de uma empresa, possui, tipicamente, 8 CPUs com 10 a 16 núcleos por processador, acrescentando-se 80 a 120 núcleos por servidor, com os servidores atuais a suportarem até 6 TB de memória [Pla]. Apesar do fenómeno de introdução de um paralelismo massivo, o disco dominou, sem dúvida, todas as otimizações de desempenho até relativamente pouco tempo atrás. É verdade que era muito lento, mas indispensável para o armazenamento da informação.

Hoje deparámo-nos com quantidades de memória principal (DRAM) totalmente superiores em servidores e, este aumento, combinado com o seu preço decrescente, despoletou a viragem para um novo paradigma no que diz respeito às bases de dados: evolução de sistemas gestores de bases de dados centradas em disco para bases de dados centradas em memória principal. Estes sistemas mantêm uma cópia primária dos seus dados em memória DRAM, enquanto que a memória persistente apenas continua a ser usada para efeitos de persistência, *backup* e recuperação da informação. [Pla]

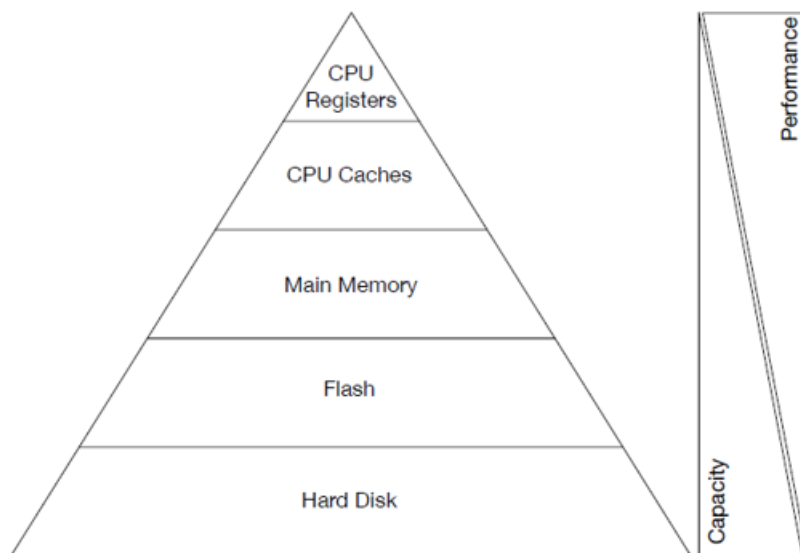


Figura 2.1: Visão concetual da hierarquia de memórias, com o seu desempenho e capacidade [Pla]

Um esquema de hierarquia de memórias pode ser visto na figura 2.1, onde é evidente a diferença em tempos de acesso a disco (memória persistente) e a memória principal (memória volátil). Só para se ter uma ideia, o tempo médio de acesso a disco está nos 10ms, enquanto que o tempo

médio de acesso a memória é 100ns [Pla]. Por outro lado, à medida que o desempenho aumenta, a capacidade diminui, o que era o principal entrave em sistemas deste tipo. O aumento desta capacidade, sobretudo, possibilitou a emergência de sistemas de bases de dados em memória RAM.

2.1.2 Codificação do dicionário

Apesar do tempo de acesso a memória ser substancialmente menor do que o tempo de acesso a disco, ainda assim, este necessita de ser minimizado o mais possível. Uma das soluções adotadas para o efeito foi usar compressão dos dados, permitindo melhores tempos de transferência entre componentes e um menor consumo de memória, pois as operações apenas são feitas com dados comprimidos, ou seja, dados representados pelo número mínimo de bits necessários.

A codificação do dicionário foi então usada, sendo um conceito relativamente simples, tanto para compreensão como para implementação. É aplicada a cada coluna individualmente, de forma a comprimir as tabelas, o que significa que cada coluna ou atributo da tabela possui o seu próprio dicionário, que contém todas as entradas diferentes para um determinado atributo. Para cada atributo, é criada uma estrutura de dados auxiliar chamada vetor de atributos, que é composto por um *ValueID*, sendo este valor mapeado para uma entrada específica do dicionário, à semelhança do que ocorre numa *hash table*. Este mecanismo revela-se muito eficiente, sobretudo, em tabelas que possuem muitos valores repetidos [Pla]. Um exemplo ilustrativo é mostrado na figura 2.2.

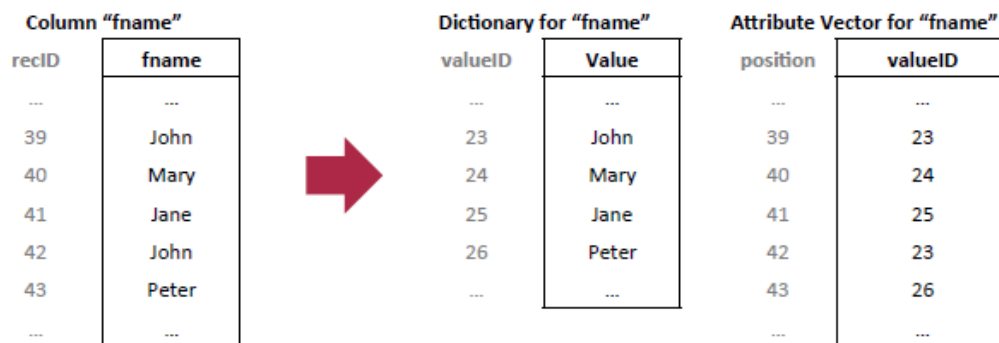


Figura 2.2: Exemplo da criação das estruturas dicionário e vetor de atributos para a coluna *fname* [Pla]

Basicamente, uma operação de pesquisa começa por verificar se o dicionário já contém o valor pedido, e, em caso afirmativo, encontra o respetivo *ValueID* associado. De seguida, é suficiente percorrer o vetor de atributos à procura deste *ValueID*, para posteriormente, no resultado da pesquisa, ser substituído o *ValueID* pelo valor correspondente no dicionário. Os dicionários são ordenados, para melhorar os tempos de acesso, após a codificação dos dados, ficando com uma complexidade de tempo logarítmica - $O(\log(n))$, utilizando uma pesquisa binária, enquanto que dicionários não ordenados têm uma complexidade de tempo linear - $O(n)$. Apesar das vantagens enumeradas, esta abordagem possui limitações, nomeadamente, quando um novo registo é inserido

na base de dados e esse valor não consta no dicionário, o que significa que o dicionário tem de ser reordenado e o vetor de atributos atualizado, o que se pode revelar um processo demorado. [Pla]

A codificação do dicionário funciona ainda como base para outras técnicas de compressão que podem ser aplicadas em colunas, como veremos na secção seguinte.

2.1.3 Técnicas de compressão

De forma a melhorar o desempenho das *queries* e diminuir o consumo de memória principal, determinadas técnicas de compressão são também aplicadas, além da codificação do dicionário. Apesar de assistirmos a uma crescente capacidade de memória principal ao longo dos últimos anos, quando se trata de processar enormes volumes de dados em memória, o processo torna-se custoso. Por conseguinte, as técnicas usadas nas aplicações empresariais devem ser leves, caso contrário, os processos de codificação e decodificação tornar-se-iam bastante penosos.

São seis as principais técnicas adotadas e mais comuns em bases de dados em memória, nomeadamente:

- *Prefix Encoding* - Usada quando uma coluna contém um valor predominante, enquanto os restantes valores raramente aparecem.
- *Run-Length Encoding* - Usada quando uma coluna possui poucos valores distintos (baixa cardinalidade) com um grande número de ocorrências.
- *Cluster Encoding* - O vetor de atributos é dividido em N blocos de tamanho fixo (geralmente 1024 elementos). Se o cluster contém apenas um único valor, este é substituído por uma única ocorrência desse valor. Caso contrário, o *cluster* permanece descomprimido.
- *Indirect Encoding* - Semelhante ao *Cluster Encoding*, esta técnica opera em blocos de dados com N elementos (tipicamente 1024). É usada quando uma tabela está ordenada por uma coluna e existe uma correlação entre as colunas. Além do dicionário global, outros, mais pequenos, são usados em cada *cluster*.
- *Sparse Encoding* - Usada quando existe muitos valores nulos ou vazios, utilizando um vetor de bits com o valor 1 para valores existentes e 0 para valores não existentes.
- *Delta Encoding* – Usada quando os dados estão ordenados e existe um grande número de valores com o mesmo prefixo. Ao contrário das técnicas apresentadas anteriormente, cujo objetivo, é reduzir o tamanho do vetor de atributos, esta aplica-se ao dicionário, tendo como missão reduzir o volume de dados no dicionário.

Das técnicas referidas, nem todas possuem um acesso direto, algo que deve ser tido em conta se estivermos perante requisitos de tempo estritos. Técnicas como *Cluster Encoding* e *Sparse Encoding* não possuem acesso direto. Contudo, todas as técnicas mencionadas alcançam o seu maior potencial quando aplicadas em tabelas ordenadas e estas apenas o podem ser através de uma única coluna. [Pla]

2.1.4 Tipos de queries

Na base da gestão do dia-a-dia de um negócio, é indispensável conhecer e fazer a distinção entre OLTP (*On-line Transactional Processing*) e OLAP (*On-line Analytical Processing*). Ao longo das últimas décadas, as aplicações empresariais poderiam ser classificadas quer como centradas em OLAP ou em OLTP. Hoje, na maioria dos casos, essa distinção não pode ser feita e espera-se que um único sistema seja capaz de lidar com ambos os tipos de *workload*. Por um lado, o OLTP é caracterizado por um grande número de pequenas transações (INSERT, UPDATE, DELETE). Estes sistemas têm como ênfase principal um processamento de consulta muito rápido, mantendo a integridade dos dados em ambientes com vários acessos e uma eficácia medida pelo número de transações por segundo. Por outro lado, o OLAP caracteriza-se por um volume relativamente baixo de transações (SELECT), embora as queries sejam, frequentemente, complexas e envolvam agregações. Para sistemas deste género, o tempo de resposta é uma medida bastante eficaz e são largamente adotados em técnicas de *Data Mining*. A tabela 2.1 resume as principais características de ambos os tipos de processamento. [Pla, Che]

Tabela 2.1: Principais diferenças entre OLTP e OLAP [Che]

Característica	OLTP	OLAP
Origem dos dados	Dados operacionais da aplicação. OLTPs são a fonte original de dados	Dados históricos provenientes de bases de dados OLTP
Propósito dos dados	Controlar e executar tarefas de negócio fundamentais	Auxiliar no planeamento, resolução de problemas e tomada de decisão
Volume	Grande número de pequenas transações	Pequeno volume de transações complexas
Velocidade de processamento	Tipicamente muito rápidas	Depende da quantidade de dados, podendo demorar muitas horas
Eficácia	Número de transações por segundo	Tempo de resposta das queries
Utilizadores	Staff comum	Gestores e Executivos

2.1.5 Layout dos dados em memória principal

Bases de dados relacionais possuem uma estrutura bidimensional, ao contrário de bases de dados em memória que possuem uma estrutura apenas de uma dimensão. Assim sendo, cabe à camada de armazenamento da base de dados decidir de que forma mapeia a informação das estruturas de tabelas bidimensionais para um espaço de endereçamento de memória linear (uma dimensão). Existem duas formas principais de representação de tabelas em memória, chamadas *layout* orientado a linha e orientado a coluna, apesar de uma combinação dos dois ser também uma opção válida (*layout* híbrido).

Não é correto afirmar que um tipo de *layout* se superioriza em relação aos outros para qualquer tipo de *workload*. Estes podem ser muito diversificados, num ambiente empresarial, de forma que encontrar o layout mais adequado é crucial para este tipo de abordagem [Pla]. Na tabela 2.2 são

Tabela 2.2: Layouts possíveis para uma abordagem em memória principal [Pla]

<i>Layout orientado a linha</i>	<i>Layout orientado a coluna</i>	<i>Layout híbrido</i>
Dados guardados em tuplos	Dados guardados orientados a atributos	Combina o melhor dos dois esquemas anteriores. Alguns atributos duma tabela podem ser guardados usando um <i>layout</i> orientado a linha, enquanto que outros são guardados num <i>layout</i> orientado a coluna
Potencia a localização de atributos para um único tuplo	Potencia a velocidade de <i>scan</i> sequencial em memória principal	
Técnicas de compressão mais fracas	Técnicas de compressão mais fortes	
Baixo custo para reconstrução, mas um custo maior para <i>scan</i> de um único atributo	Reconstrução de tuplos é custosa	

enumeradas as principais diferenças entre estes três tipos de armazenamento dos dados. Na figura 2.3 são esquematizadas as operações em linha e coluna para os três tipos de organização de dados.

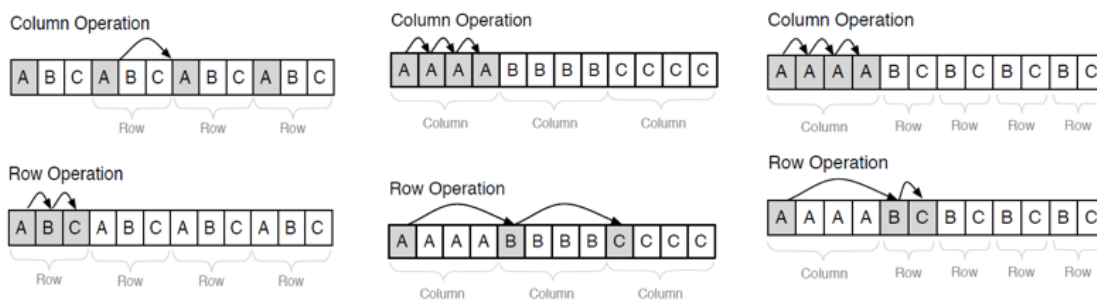


Figura 2.3: Operações em linha e coluna para ambos os tipos de *layout*: orientado a linha, orientado a coluna e híbrido, da esquerda para a direita [Pla]

2.1.6 Operações da base de dados

2.1.6.1 Insert

Quando falamos em bases de dados em memória, as suas principais operações merecem destaque devido às particularidades inerentes ao sistema, nomeadamente o seu formato de armazenamento em coluna e a codificação do dicionário. No que concerne à inserção de um novo registo na base de dados, a forma como funciona depende do formato de armazenamento adotado.

Para um formato de armazenamento orientado a linha o novo tuplo é anexado no fim da tabela, enquanto que num armazenamento orientado a coluna (o SAP HANA é maioritariamente organizado desta forma), se o novo registo não necessitar de uma nova entrada no dicionário, isto porque o seu valor já está lá presente, o *ValueID* é anexado ao vetor de atributos. Caso contrário, uma nova entrada no dicionário é necessária, o dicionário tem de ser reordenado caso a inserção do novo valor afete a ordenação, e o vetor de atributos tem de ser atualizado, o que se revela custoso em tempo e recursos. [Pla]

2.1.6.2 Update

O processo de update envolve a respetiva atualização do dicionário, a sua reorganização, a reorganização do vetor de atributos e a correspondente atualização de todos os valores antigos pelos valores recentes no vetor. É um processo custoso, principalmente quando se trata de atualizar um único valor. De forma a evitar todo este processo, o SAP HANA faz uso de uma abordagem inovadora, chamada *Insert-Only*, que será falada mais à frente. [Pla]

2.1.6.3 Delete

Esta operação termina a validade de um tuplo, e a informação que um determinado tuplo deixou de ser válido é armazenada na base de dados. Existem dois tipos de *delete*: pode ser físico ou lógico. No primeiro caso, o tuplo é removido da base de dados de forma que deixa de ser possível aceder-lhe fisicamente, ou seja, o registo é eliminado completamente da base de dados, o que pode acontecer, por exemplo, devido a questões legais. O *delete* lógico apenas termina a validade de um item no *dataset*, e não remove a informação da base de dados, mas estabelece uma *flag* no registo indicando que o mesmo não é válido, apesar do tuplo continuar a estar disponível para *queries* temporais, como por exemplo, consulta do histórico de encomendas de um determinado cliente. [Pla]

2.1.6.4 Select

Uma vez que o SQL apresenta uma descrição declarativa do resultado obtido da base de dados, é necessário um conjunto ordenado de passos de execução para extrair os dados da base de dados, chamado plano de execução da *query*. Para cada *query* podem existir múltiplos planos de execução que conduzem ao mesmo resultado, mas com desempenhos diferentes. Os otimizadores da base de dados são usados para calcular o custo de diferentes planos de execução. O objetivo principal é reduzir ao máximo o tamanho do conjunto de resultados o mais cedo possível e, para isso, deve-se aplicar seleções o mais cedo possível, ordenar seleções sequenciais de forma que as mais restritivas são executadas em primeiro lugar e ordenar junções correspondentes às cardinalidades das suas tabelas (tabelas mais pequenas são usadas primeiro). Por exemplo, se existe uma coluna com baixa cardinalidade, esta é aquela que deve ser selecionada primeiro. [Pla]

O formato de armazenamento é algo muito importante a considerar. Se o objetivo da consulta é obter tuplos completos, é recomendado usar um *layout* orientado a linha, por outro lado, se queremos obter várias entradas com o mesmo atributo já faz mais sentido um *layout* orientado a coluna. Uma *query* simples tem desempenho melhor num formato orientado a linha, apesar do tempo de resposta num formato de coluna ser bastante aceitável, enquanto que numa *query* complexa verifica-se o oposto, apesar de um formato em linha poder não devolver o resultado em tempo real. Desta forma, pode-se concluir que estes formatos diferentes devem ser usados em circunstâncias diferentes, de forma a maximizar o desempenho da aplicação. [Pla]

2.1.6.5 Abordagem *Insert-Only*

Os dados guardados numa base dados alteram-se ao longo do tempo e essas alterações devem ser rastreáveis, uma vez que, por questões legais (por exemplo, auditorias financeiras) ou necessidade própria, as empresas podem pretender aceder aos seus dados históricos.

Utilizando esta abordagem, as aplicações não realizam qualquer tipo de *updates* ou *deletes* nos dados existentes armazenados fisicamente, mas criam novos tuplos e gerem a validade dos mesmos. Desta forma, é também possível verificar de que modo determinada informação evoluiu com o tempo, sem haver necessidade de limpar dicionários, o que se revela custoso em consumo de memória. [Pla]

Entre os principais benefícios desta abordagem, destacam-se os seguintes:

- Facilita a execução de *time-travel queries*, o que permite aos utilizadores verem a informação como ela era num ponto anterior do tempo. Este acesso facilitado permite à gestão da empresa eficientemente analisar o seu desenvolvimento e, assim, conduzir a boas decisões estratégicas;
- Simplifica a implementação de mecanismos de paralelismo, por exemplo, controlo de concorrência de multi-versões;
- Para bases de dados em memória, esta abordagem elimina a necessidade de limpeza de dicionários em operações de *update*.

Existem duas formas de implementação desta abordagem: representação por pontos e representação por intervalos. Na primeira, para determinar a validade dos tuplos, um único campo (*valid from*) é usado para guardar a data de inserção do tuplo, enquanto que na segunda, para determinar a validade dos tuplos, são usadas duas colunas (*valid from* e *valid to*) que contêm informação acerca do intervalo de tempo no qual o tuplo é considerado válido. [Pla]

2.1.7 Paralelismo

De forma a acelerar o processamento das *queries* é necessário pensar em paralelismo. A sua tendência é aumentar ao longo dos anos, fruto da evolução constante sofrida pelo *hardware*, com um crescimento evidente no número de núcleos dos processadores. Este ambiente é essencial às bases de dados em memória, uma vez que maximizam o uso de paralelismo para obter o melhor desempenho possível.

Numa base de dados em memória, são usados dois tipos de paralelismo: paralelismo em *pipeline* e paralelismo de dados. No primeiro processo, o operador seguinte inicia-se enquanto o atual ainda não terminou, mas já produziu resultados parciais e, assim, o tempo de execução dos operadores sobrepõem-se parcialmente. Esta cascata de operações é denotada como *pipelining* e pode ser estendida sem nenhuns limites. No paralelismo de dados, o conjunto de dados é particionado de modo que os operadores de uma *query* trabalhem em partes individuais do conjunto de dados em paralelo. Consequentemente, os resultados das operações em paralelo são fundidos no conjunto de resultados completo [Pla], como se pode ver no esquema da figura 2.4.

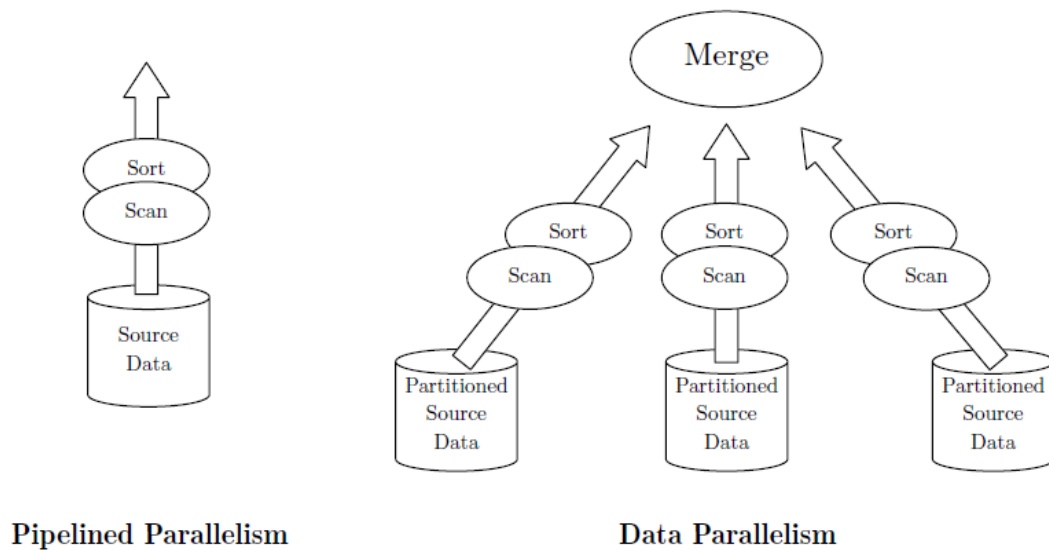


Figura 2.4: Tipos de paralelismo em bases de dados em memória principal [Pla]

2.1.8 Cache de agregação

Sistemas OLAP e OLTP empregam abordagens diferentes quando lidam com *queries* de agregação. Por um lado, os sistemas OLAP fazem grande uso de vistas materializadas, por outro, a manipulação de agregações em sistemas OLTP é, geralmente, feita dentro da aplicação, mantendo algumas tabelas sumárias pré-definidas.

Com a tendência crescente de utilização de bases de dados em memória, com um formato de armazenamento em coluna, deixa de ser necessário haver uma separação entre OLTP e OLAP, uma vez que estas plataformas são capazes de lidar com *workloads* mistos, com ambas as *queries*, analíticas e transacionais, a coexistirem num único sistema. Apesar das capacidades de agregação das bases de dados em memória, o acesso a tuplos de um agregado materializado (vista materializada cuja consulta contém funções de agregação) verifica-se sempre mais rápido do que agregação *on-the-fly*. Enquanto que estratégias de manutenção de vistas materializadas existentes são aplicáveis em sistemas de bases de dados em memória, a sua arquitetura é bastante apropriada para uma estratégia de cache de *queries* de agregação e uma aplicação de técnicas de manutenção de vista incremental.

Num sistema gestor de base de dados em memória, o armazenamento pode ser separado em dois componentes: o armazenamento principal, otimizado para leitura, e o *differential buffer*, otimizado para escrita. Novos registos são inseridos sempre no *differential buffer* e, periodicamente, são fundidos para o armazenamento principal (operação de *merge*). Adotando esta arquitetura, os agregados materializados não necessitam de ser invalidados quando novos registos são inseridos no *differential buffer*, uma vez que estes apenas são definidos em registos do armazenamento principal. [Pla]

2.1.9 Differential Buffer

O conceito de *differential buffer*, também conhecido como *delta buffer* ou *delta store* divide a base de dados em dois componentes, como referido na secção anterior. Todas as operações de *insert*, *update* e *delete* são realizadas nesta estrutura juntamente com vetores de validade. Desta forma, todas as modificações aos dados ocorrem apenas nesta zona de armazenamento, excluindo o armazenamento principal destas tarefas.

O estado atual dos dados é a conjunção do *differential buffer* e do armazenamento principal, o que significa que cada operação de leitura, deve ser efetuada em ambos os componentes de armazenamento. Uma vez que o tamanho do *differential buffer* é substancialmente mais pequeno que o armazenamento principal, esta leitura tem um pequeno impacto no desempenho, visto o *differential buffer* ser otimizado para escrita e não para leitura, ao contrário do que acontece com a zona principal.

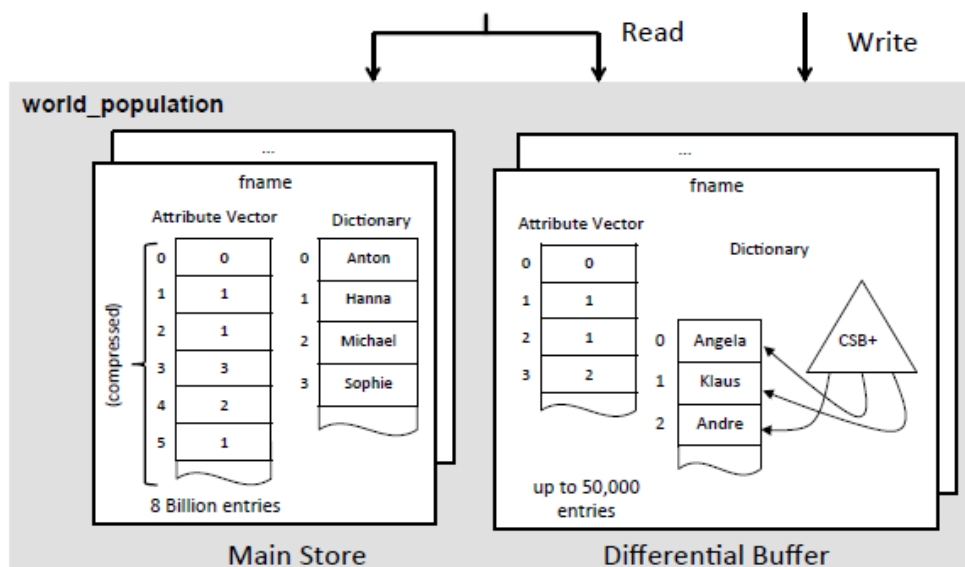


Figura 2.5: Conceito de *differential buffer* [Pla]

Durante a execução das *queries*, estas são logicamente divididas no armazenamento principal comprimido e no *differential buffer*. Após os resultados de ambas as *subqueries* serem devolvidos, as representações intermédias são combinadas para construir um resultado totalmente válido do estado de dados atual [Pla]. Um esquema deste conceito pode ser visualizado na figura 2.5.

2.2 Plataformas existentes

De seguida são apresentadas soluções em memória já comercializáveis, com especial ênfase para a plataforma SAP HANA, utilizada no desenvolvimento da aplicação. Foi selecionada uma ferramenta de três das maiores fabricantes mundiais de *software* e bases de dados, Microsoft, Oracle e SAP. Contudo, nos dias de hoje, a maioria dos principais fabricantes de base de dados já se

precaveu e possui a sua própria solução em memória principal [LL16]. Estas soluções inovadoras contrastam com sistemas de memória persistente, como é o caso do SAP R/3, utilizado também no âmbito deste trabalho.

A *Oracle Database 12c*, apesar de não ter sido utilizada nos principais estudos investigados, possui um design e uma abordagem muito semelhante ao SAP HANA, e por essa razão, mereceu ser considerada. O *Hekaton* surge como uma tecnologia recente, integrada na versão tradicional da solução Microsoft e, foi escolhida, por ser um sistema ligeiramente diferente dos outros dois apresentados. O SAP HANA é a base de dados em memória mais adotada a nível mundial e a mais utilizada nos estudos investigados. Embora o SAP R/3 seja um sistema de memória persistente, este merece destaque, uma vez que foi utilizado no âmbito desta dissertação.

2.2.1 Microsoft SQL Server 2014 (Hekaton)

Dentro das soluções de bases de dados em memória, o SAP HANA aparece como a plataforma com maior reconhecimento, apesar de existirem muitas possibilidades para as organizações que procuram uma opção em memória para as suas aplicações de bases de dados [Cur, LL16]. A gigante Microsoft detém um *software* que se classifica como sendo uma espécie de plataforma de computação em memória. E espécie pois os próprios consideram-no um sistema OLTP em memória, que é um motor de base de dados, chamado *Hekaton*, otimizado em memória, totalmente integrado num motor SQL Server otimizado para OLTP. Foi lançado na versão SQL Server 2014. [Cur, LL16]

O facto do SQL Server 2014 e o *Hekaton* não serem sistemas distintos permite aos utilizadores declararem uma ou mais tabelas numa base de dados otimizada em memória. Esta abordagem oferece várias vantagens: evita o aborrecimento e despesa de possuir outro SGBD; apenas as tabelas mais críticas em desempenho necessitam de estar em memória, todas as outras permanecem inalteradas e esta conversão pode ser feita gradualmente, ou seja, uma tabela ou *stored procedure* de cada vez.

O motor *Hekaton* revela uma melhoria em mais do que uma ordem de grandeza, em eficiência e escalabilidade, com alterações mínimas e incrementais às aplicações dos utilizadores [DFI+13]. OLTP refere-se ao processamento transaccional online, enquanto que OLAP diz respeito ao processamento analítico, como referido anteriormente.

2.2.2 Oracle Database 12c

Esta solução da Oracle tem uma abordagem já muito semelhante ao SAP HANA. É, de acordo com a multinacional dos EUA, desenhada e pensada para executar tanto OLTP como OLAP a partir da mesma base de dados, o que aumenta o desempenho e potencia as suas capacidades de duas formas:

- Por um lado, o facto de apenas haver uma base de dados elimina a necessidade de mover todos os dados de uma base de dados (ou parte deles) para outra antes da análise poder ser realizada.

- Por outro lado, uma vez que a base de dados é a mesma, as *queries* podem ser executadas no *dataset* completo em qualquer altura.

A capacidade para realizar estas *ad-hoc queries* tem sido uma das maiores preocupações dos administradores de bases de dados. O objetivo da integração do armazenamento de dados orientado a coluna, em memória, com a base de dados é melhorar o desempenho dos *workloads* analíticos sem comprometer o desempenho das transações que continuam a usar o formato de armazenamento tradicional da Oracle (orientado a linha) em memória. [Cur]

Esta base de dados de dois formatos utiliza-os com propósitos diferentes: o formato em coluna visa acelerar as *queries* analíticas e o formato em linha (tradicional) adequa-se melhor para *queries* transacionais. Um armazenamento dedicado em coluna, chamado *In-Memory Area* é responsável pelo volume de dados em coluna, onde cada uma pode ser comprimida a diferentes níveis de compressão. Esta área representa um sub-conjunto da área global partilhada da base de dados (SGA). [CJG15]

2.2.3 SAP HANA

Desenvolvida, em 2010, pela empresa alemã SAP, especializada em *software* empresarial, a plataforma SAP HANA foi pensada para responder a qualquer tipo de *query* em tempo real. A base de dados HANA posiciona-se como o componente principal dentro da ferramenta SAP HANA, de forma a suportar processos de negócio analíticos e bem complexos juntamente com *workloads* transacionais e operacionais [FCP⁺12, SFL⁺12]. Para se ter uma noção, no terceiro trimestre de 2015 o número de clientes a usarem SAP HANA ultrapassou os 7200, enquanto no ano anterior esse número era cerca de metade. [Den]

2.2.3.1 Características diferenciadoras

O SAP HANA possui algumas características chave que o distinguem da concorrência e de outros sistemas de gestão de bases de dados baseados em SQL. Estas representam os pilares da filosofia que está por trás da base de dados HANA e merecem realce as seguintes:

- Ambiente de processamento de *queries multi-engine*: de forma a suportar as funcionalidades nucleares das aplicações empresariais, a base de dados SAP HANA fornece um acesso baseado em SQL a dados estruturados relacionalmente com total suporte transacional. Fornece ainda um motor de pesquisa de texto além do motor de *queries* relacional clássico. Por último, um motor de grafos fornece a capacidade para correr algoritmos de grafos em redes de entidades de dados para suportar as aplicações de negócio.
- Representação de objetos de negócio específicos da aplicação: a base de dados HANA é capaz de fornecer uma compreensão profunda dos objetos de negócio usados na camada da aplicação. A plataforma também fornece acesso a lógicas de negócio específicas que estão implementadas diretamente no motor da base de dados.

- Exploração dos desenvolvimentos no *hardware*: nomeadamente o aumento considerável da capacidade de memória principal, os números de núcleos por processador (nó), configurações de clusters e as características de armazenamento da memória flash e SSD, de forma a usufruir eficazmente dos recursos de *hardware* e garantir um bom desempenho na execução das *queries*. A base de dados SAP HANA foi construída de raiz para executar em ambientes centrados em memória principal e em paralelo.
- Comunicação eficiente com a camada da aplicação: os planos no desenvolvimento do SAP HANA foram, por um lado, fornecer uma comunicação de memória partilhada com servidores de aplicações proprietárias da SAP e alinhar os tipos de dados usados em cada um. Por outro lado, foi planeado integrar a tecnologia do servidor da aplicação diretamente na infraestrutura do *cluster* da base de dados, o que permite uma execução entrelaçada da lógica da aplicação e da gestão da base de dados. [FCP⁺12]

2.2.3.2 Arquitetura geral

A base de dados SAP HANA é um sistema de gestão de dados centrado em memória que potencia as capacidades do *hardware* moderno, nomeadamente, uma enorme capacidade de memória principal e processadores com vários núcleos [FCP⁺12]. Este sistema fornece uma infraestrutura para aplicações empresariais SAP atuais e futuras. O seu design e arquitetura foram pensados com base nas alterações significativas nos requisitos que ocorrem na gestão de dados, nas aplicações empresariais modernas, assim como, e nos desenvolvimentos recentes nas arquiteturas de hardware. [BTMF12]

Conforme se pode ver na figura 2.6, o sistema possui um componente de Conexão e Gestão de Sessões que cria e gere sessões e conexões para os clientes da base de dados. Uma vez estabelecida a sessão, os clientes podem usar SQL, SQL Script, MDX ou outra linguagem específica de domínio como, por exemplo, a linguagem proprietária da SAP (FOX) para aplicações de planeamento, para comunicarem com a base de dados HANA. Esta plataforma garante transações ACID (Atomicidade, Consistência, Isolamento, Durabilidade), onde o Gestor de Transações coordena as transações da base de dados, controla o isolamento transacional e mantém rastreio das transações em execução e terminadas. Para controlo de concorrência, o SAP HANA implementa o princípio MVCC, que permite a leitura de transações de longa execução sem bloquear transações de atualização. O MVCC, juntamente com o mecanismo de *time-travel*, permite a execução de *queries* temporais dentro do Motor Relacional. O SQL Script e as linguagens suportadas são traduzidas pelos seus compiladores específicos numa representação interna chamada Modelo de Cálculo. A execução destes modelos fica a cargo do Motor de Cálculo. A base de dados possui três motores de armazenamento em memória: um Motor Relacional, um Motor de Grafos e um Motor de Texto.

O primeiro suporta representações físicas orientadas a linha e orientadas a coluna de tabelas relacionais. Dados orientados a coluna são guardados num formato altamente comprimido, visando uma melhoria na eficiência do uso de memória e uma aceleração na transferência de dados de armazenamento para memória ou de memória para CPU. As tabelas da base de dados,

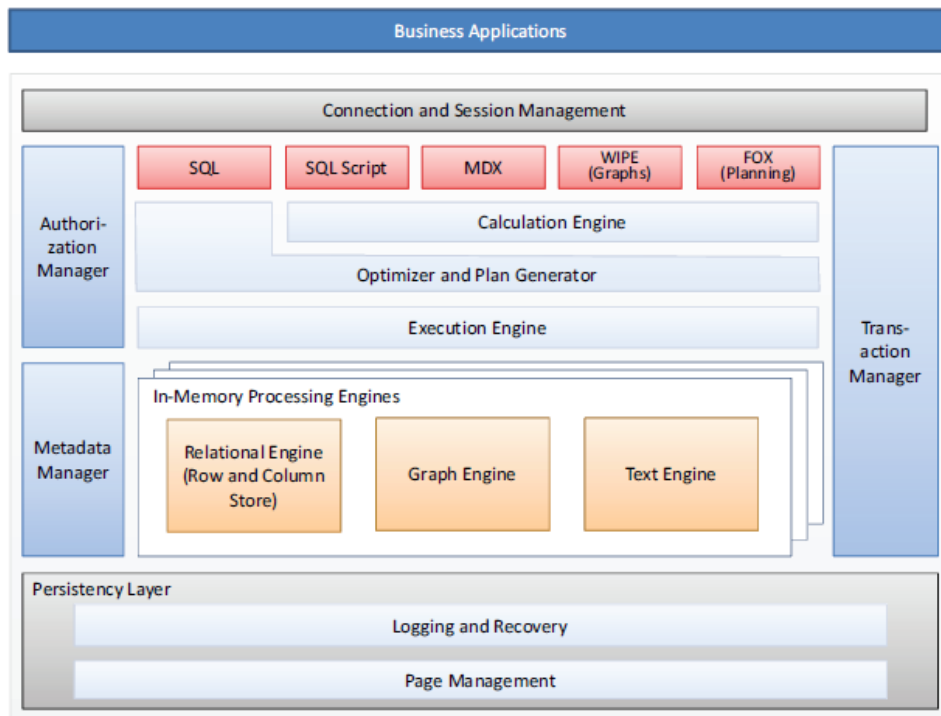


Figura 2.6: Arquitetura da base de dados HANA [FCP⁺12]

em ambos os formatos anteriormente mencionados, podem ser perfeitamente combinadas numa declaração SQL e, conseqüentemente, as tabelas podem ser movidas de uma forma de representação para a outra. O Motor de Grafo suporta uma representação e um processamento eficiente de grafos de dados com um sistema de escrita flexível e encontra-se posicionado para, de uma forma ótima, suportar aplicações de planejamento de recursos com enormes números de recursos individuais e interdependências complexas. O Motor de Texto fornece capacidades de indexação e pesquisa de texto, tais como pesquisa exata de palavras e frases, *fuzzy search* que tolera erros de escrita e pesquisa linguística que encontra variações de palavras baseadas em regras linguísticas. [FCP⁺12, BTMF12]

2.2.4 SAP R/3

O sistema SAP R/3 é considerado a terceira geração de módulos de *software* totalmente integrados que realizam determinadas funções de negócio, baseando-se em práticas líderes multinacionais. Foi lançado em 1992 mas até então sofreu variadas alterações de forma a melhorar aspetos menos conseguidos da plataforma.

Foi desenvolvido na linguagem ABAP/4, a linguagem mais utilizada neste sistema. A sua arquitetura é baseada num modelo cliente-servidor formado por duas ou três camadas, embora o último seja o recomendado. A camada de apresentação é responsável pela interface gráfica do sistema, enquanto que na camada da aplicação é onde são executados as aplicações em ABAP/4. Por

fim, a camada de gestão de dados (bases de dados relacionais) tem como função o armazenamento de toda a informação, incluindo os próprios programas.

Este sistema oferece um conjunto de aplicações integradas compreensivas que vão de encontro às necessidades das organizações e, desta forma, foi líder de mercado para sistemas integrados de administração de empresas. Apenas existe uma única base de dados e todos os componentes se ligam a ela. Flexibilidade, robustez e escalabilidade foram outros aspetos tidos em conta aquando da implementação do SAP R/3.

Um dos principais problemas que se verificaram, ao longo dos últimos anos, foi o facto da codificação neste sistema se ter tornado complexa, assim como, a sua arquitetura, aliados ao suporte praticamente inexistente para outras linguagens. [Bun, KKZ99]

2.3 Vantagem competitiva e conclusões

Uma abordagem em memória principal, como o SAP HANA, apresenta variadas vantagens ao negócio de uma organização conforme pode ser visto no esquema da figura 2.7. As suas características específicas conduzem a determinados efeitos, de primeira e segunda ordem, como redução dos tempos de consulta e processamento de grandes e complexos volumes de informação até uma análise avançada de negócio e a capacidade de lidar com ambos os tipos de processamento, analítico e transacional, no mesmo sistema. [vBDMU13]

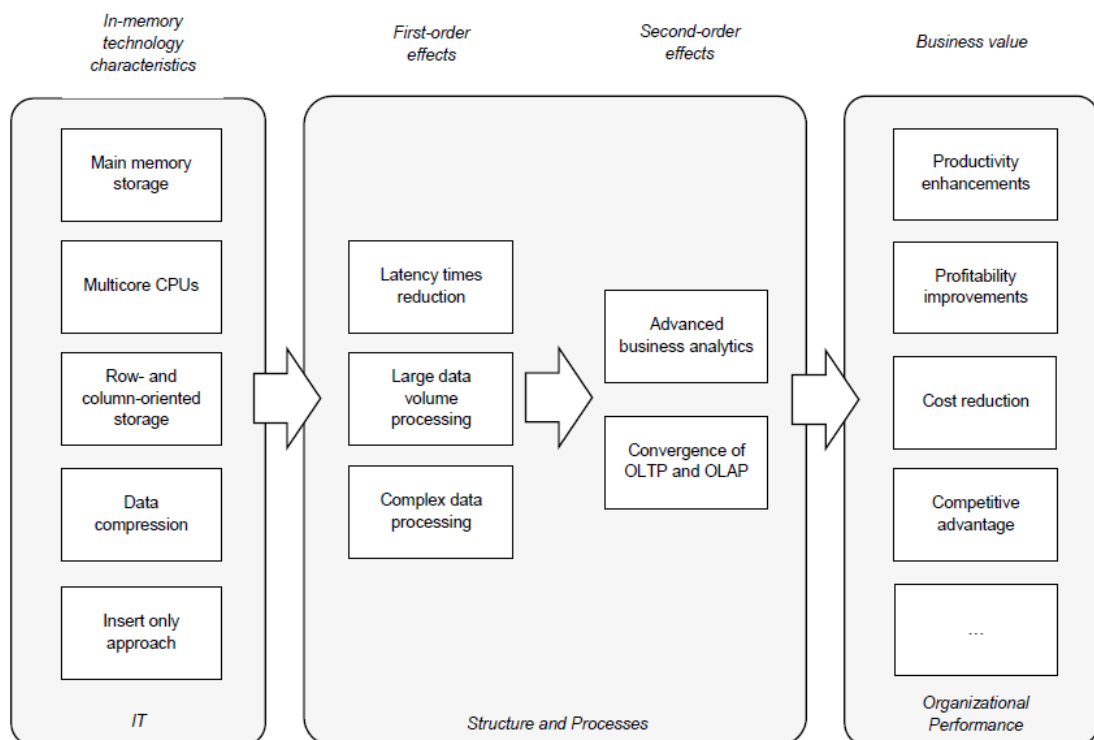


Figura 2.7: Criação de valor através duma tecnologia em memória DRAM [vBDRM13]

Computação em memória principal

A importância da tomada de decisões, num ambiente empresarial, revela-se um dos principais fatores de sucesso e criação de vantagem competitiva para o negócio. As bases de dados em memória principal, como a HANA, possuem características únicas que permitem potenciar a criação de valor, através de melhorias da produtividade, da rentabilidade, redução de custos ou melhoria na gestão do inventário. [vBDMU13]

O objetivo é fazer uso destas mesmas características de forma a otimizar, ao máximo, os tempos de resposta dos relatórios usados pela empresa de construção para a gestão do seu inventário, comparando-os com os tempos obtidos atualmente, no sistema SAP R/3. A questão que se coloca é se fará sentido para a organização rentabilizar este investimento para resolver problemas associados à gestão de inventário e flexibilizar a tomada de decisões, algo ainda não muito explorado, apesar de estudos apontarem esta melhoria de inventário como um dos aspetos de criação de vantagem competitiva, conforme pode ser visto no capítulo seguinte.

Capítulo 3

Revisão da Literatura

A revisão da literatura foi efetuada com base, maioritariamente, em bases de dados bibliográficas especializadas no domínio. Foram consultadas a *ACM Digital Library*, a *Scopus* e a *Web of Science*. Como pesquisa secundária foram ainda consultados o *Google Scholar*, assim como, algumas páginas *web*. O intervalo de tempo considerado para a pesquisa foram os últimos 10 anos.

Palavras-chave usadas na pesquisa foram: *in-memory computing*, *SAP HANA*, *Big Data*, *shop floor data*, *value creation*, *inventory management* e *competitive advantage*. Na tabela 3.1 podem ser consultados todos os recursos utilizados na revisão da literatura e na elaboração do estado da arte.

Tabela 3.1: Revisão da Literatura

Título	Autor(es)	Ano	Resumo
<i>Efficient Transaction Processing in SAP HANA Database – The End of a Column Store Myth [SFL⁺12]</i>	Vishal Sikka Franz Färber Wolfgang Lehner Sang Kyun Cha Thomas Peh Christof Bornhövd	2012	São apresentadas as principais funcionalidades que diferenciam a base de dados SAP HANA dos motores de bases de dados relacionais tradicionais. É esboçada a arquitetura geral e os critérios de desenho do sistema. O objetivo é mostrar de que forma a base de dados HANA é capaz de, eficientemente, trabalhar em ambos os tipos de workloads
<i>SAP HANA: The Evolution from a Modern Main-Memory Data Platform to an Enterprise Application Platform [SFGL13]</i>	Vishal Sikka Franz Färber Anil Goel Wolfgang Lehner	2013	Este artigo esboça a visão e o próximo passo na evolução do SAP HANA, desde uma plataforma de dados principais numa plataforma de aplicações empresariais
<i>Modern Main-Memory Database Systems [LL16]</i>	Per-Ake Larson Justin Levandoski	2016	Fornecer uma visão geral de desenvolvimentos recentes em sistemas de bases de dados em memória. São apresentadas as principais características deste tipo de sistemas, que permitem alcançar o excelente desempenho destas BD. O artigo foca-se, sobretudo, em questões de desenho e arquiteturas que devem ser tidas em conta aquando da criação de um sistema deste género

Revisão da Literatura

Título	Autor(es)	Ano	Resumo
<i>Hekaton: SQL Server's Memory-Optimized OLTP Engine [DF⁺13]</i>	Cristian Diaconu Craig Freedman Erik Ismert Per-Ake Larson Pravin Mittal Ryan Stonecipher Nitin Verma Mike Zwilling	2013	O artigo fornece uma visão geral dos princípios de design do motor Hekaton e descreve alguns resultados experimentais do sistema
<i>SAP HANA Database – Data Management for Modern Business Applications [FCP⁺12]</i>	Franz Färber Sang Kyun Cha Jürgen Primsch Christof Bornhövd Stefan Sigg Wolfgang Lehner	2011	São apresentadas as principais características do SAP HANA, realçando os aspetos diferenciadores em relação a plataformas tradicionais. São referidas as vantagens e a arquitetura do sistema
<i>Data Management with SAPs In-Memory Computing Engine [BTMF12]</i>	Joos-Hendrik Boese Cafer Tosun Christian Mathis Franz Faerber	2012	São apresentadas introspeções arquiteturais e tecnológicas da base de dados SAP HANA, assim como desafios de investigação para o desenvolvimento de aplicações empresariais futuras
<i>In-Memory Database Systems - A Paradigm Shift [GVV13]</i>	Mohit Kumar Gupta Vishal Verma Megha Singh Verma	2013	Este artigo explora uma abordagem em memória, assim como os problemas e os desafios associados. Também apresenta algumas soluções de sistemas deste género disponíveis no mercado
<i>A Common Database Approach for OLTP and OLAP Using an In-Memory Column Database [Pla09]</i>	Hasso Plattner	2009	Foca-se nos formatos de armazenamento no que diz respeito a questões de desempenho. Refere uma abordagem inovadora – Insert-Only - e o impacto no desenvolvimento de aplicações
<i>Study of SAP HANA in the in-memory context [BDMB16]</i>	Gabriel Borges	2016	O foco da dissertação centra-se no valor para o negócio de uma base de dados em memória principal, através da comparação de tempos de execução de queries, analíticas e transacionais, no SAP HANA e numa base de dados em memória persistente, o SAP R/3
<i>How In-memory Technology Can Create Business Value: Insights from the Hilti Case [vBDRM13]</i>	Jan vom Brocke Stefan Debortoli Oliver Müller Nadine Reuter	2014	São apresentados cenários possíveis, devido ao elevado poder de computação, de uma abordagem em memória (SAP HANA) e são identificados princípios de criação de valor para o negócio através desta tecnologia
<i>Data-intensive applications, challenges, techniques and technologies: A survey on Big Data [PCZ14]</i>	Philip Chen Chun-Yang Zhang	2014	Este artigo visa demonstrar uma visão mais detalhada do problema de Big Data, incluindo aplicações de Big Data, oportunidades e desafios, assim como técnicas e tecnologias que são adotadas atualmente. São enumeradas as principais vantagens e ajudas que pode trazer para as empresas, com face num estudo
<i>SAP HANA: Not Only In-memory Game In Town [Cur]</i>	Curtis Franklin Jr.	2015	Apresenta alternativas, existentes ao SAP HANA, nomeadamente no que diz respeito aos maiores implementadores mundiais de bases de dados, como a Oracle e a Microsoft, descrevendo aspetos destes sistemas
<i>Importance of Decision Making in Management [Akr]</i>	Gaurav Akrani	2011	São apresentadas as principais vantagens da tomada de decisões na gestão de um determinado negócio
<i>A Data Mining Experiment on Manufacturing Shop Floor Data [JKLS07]</i>	J. Jenkole P. Kralj N. Lavrac A. Sluga	2007	O artigo descreve uma experiência de data mining realizada num problema do mundo real. O objetivo é examinar a utilidade da experiência e visualizar o shop floor data, de forma a suportar a tomada de decisões numa empresa de manufatura
<i>A Requirement for Traceability of Production Logs in Large-scale Shop Floor Data [PC15]</i>	Jaehui Park Su-young Chi	2015	O objetivo é desenvolver um sistema de informação para manufatura preditiva, que pode capturar potenciais fatores de risco, tais como, o progresso das máquinas e a tendência de perda de tempos de produção
<i>In-Memory Database Business Value: Results from a Study on Retail Innovation [vBDMU13]</i>	Jan vom Brocke Stefan Debortoli Oliver Müller Axel Uhl	2013	Foca-se num estudo explorativo entre CIO's e especialistas de tecnologia da indústria de retalho. Concentra-se na criação de valor para o negócio em geral e apresenta cinco cenários de como potenciar esta tecnologia na indústria de retalho. Pode-se aplicar a outras áreas de negócio

Revisão da Literatura

<i>Manufacturing Software Managing Factory Data [Wau]</i>	Patrick Waurzyniak	2012	Descreve ferramentas e técnicas eficientes de gestão de dados que potencializam métricas críticas da área da fábrica (shop floor)
<i>SAP R/3 Overview [Bun]</i>	Bunty Jain	2012	Apresenta as principais características do ERP SAP R/3, descrevendo aspetos referentes à sua arquitetura e design. Também descreve as tecnologias utilizadas
<i>SAP's Q2 FY2015 – more color on the results, S/4 is a'coming, HANA revived [Den]</i>	Den Howlett	2015	Entrevista a um dos diretores executivos da SAP, Rob Enslin, acerca do sucesso da base de dados em memória HANA e a da estratégia adotada pela empresa alemã
<i>Query Optimization in Oracle 12c Database In-Memory [CJG15]</i>	Dinesh Das Jiaqi Yan Mohamed Zait Satyanarayana Valluri Nirav Vyas Ramurajan Krishnamachari Prashant Gaharwar Jesse Kamp Niloy Mukherjee	2015	O artigo descreve as alterações feitas ao otimizador das <i>queries</i> , de forma a gerar planos de execução otimizados para um determinado formato. Refere características deste sistema e como o otimizador contribui para o elevado desempenho desta base de dados
<i>Performance Tuning for SAP R/3 [KKZ99]</i>	Alfons Kemper Donald Kossmann Bernhard Zeller	1999	O artigo fornece uma visão global do ERP largamente adotado SAP R/3 e refere vários aspetos do desempenho deste sistema. São apresentadas experiências conduzidas, tanto em termos de processamento transacional (OLTP) como analítico (OLAP)
<i>Feel the Power: Big Data, Big Opportunity - Is Your Commodity Value Chain Smart? [SM06]</i>	Micheal Swartz Shobhit Mathur	2006	Este documento apresenta um exemplo concreto da possível criação de valor e melhoria da gestão do negócio na indústria de petróleo e gás, ao adotar uma posição e estratégia capaz de lidar com <i>Big Data</i>
<i>Big Data: The Management Revolution [MB]</i>	Andrew McAfee Erik Brynjolfsson	2012	O documento apresenta a forma de exploração de <i>Big Data</i> , de forma a melhorar o desempenho das empresas. Refere a importância de uma tomada de decisões com base nesta informação, assim como, as áreas importantes no processo de transição para uma estratégia de <i>Big Data</i>

3.1 Discussão crítica

De acordo com a revisão da literatura realizada foram descobertos alguns estudos e investigações efetuados, tendo como base um sistema de computação em memória, nomeadamente, uma base de dados em memória e a possível criação de valor associada ao uso destes sistemas.

Em [vBDRM13], estudo realizado em parceria com a *Hilti Corporation*, foi utilizado SAP HANA de forma a despoletar e mostrar as vantagens na criação de valor e correspondente vantagem competitiva para o negócio, entre elas uma melhor gestão de inventário da organização. Outro estudo, aplicado à indústria do retalho, mostra de que forma uma tecnologia em memória contribui para a criação de valor e apresenta diferentes cenários de como potenciar esta tecnologia inovadora nesta indústria [vBDMU13]. Estes cenários podem ser transpostos para outras áreas de negócio, como por exemplo, um preço dinâmico, onde a ideia é, regularmente, ajustar os preços dos bens a vender dependendo de fatores como, níveis de inventário em tempo real, a procura atual ou a qualidade dos bens perecíveis. Para facilitar este cenário, pode ser usada uma base de dados em memória, que é capaz de gerir os níveis de inventário em tempo real ou perto disso. Importa salientar que não foi utilizada nenhuma plataforma em concreto, mas foram referidas possíveis vantagens para o negócio em utilizar um sistema com uma abordagem em memória RAM.

Big Data foi uma das palavras-chave usadas na pesquisa e os recursos [PCZ14] e [MB] pretendem demonstrar os principais desafios e oportunidades deste fenómeno, a importância na análise de tais dados para as organizações e tecnologias e técnicas adotadas atualmente. Foi essencial compreender as possíveis vantagens para o negócio tendo em conta uma estratégia de *Big Data* e um exemplo concreto pode ser visto em [SM06], onde são referidos os aspetos de melhoria

da gestão e criação de valor, aplicados à indústria de petróleo e gás.

Informação acerca da base de dados HANA e da sua arquitetura foi importante recolher visto ser usada na implementação da aplicação otimizada, assim como a sua evolução ao longo do tempo. Para isso foram consultados recursos como [SFL⁺12], [SFGL13], [FCP⁺12] e [BTMF12]. Foi também relevante conhecer outras plataformas em memória já comercializáveis ([LL16], [DFI⁺13], [GVV13], [CJG15] e [Cur]). Apesar do foco deste trabalho ser bases de dados em memória, nomeadamente, a base de dados HANA, foi importante conhecer um pouco acerca do sistema SAP R/3, base de dados centrada em disco, uma vez que foi usada no âmbito da dissertação, na análise do programa do cliente. Para tal contribuíram [KKZ99] e [Bun].

De forma a suportar a tomada de decisões em organizações, uma análise em tempo real ou perto disso é mandatória. Esta tomada de decisões assume um papel fulcral na gestão de um negócio ([Akr]) e torna a computação em memória num assunto com extrema importância na criação de vantagem competitiva sustentável. Em [JKLS07] é feita uma experimentação de *data mining*, cujo objetivo é visualizar o *shop floor data*, suportando a tomada de decisões numa empresa de manufatura. Em [PC15] o objetivo foi desenvolver um sistema de informação para análise preditiva em *shop floor data*, que permite identificar potenciais fatores de risco, desde o progresso das máquinas aos tempos de perda no processo de produção.

Um estudo de dados no contexto do SAP HANA foi efetuado em [BDMB16] e, teve como objetivo, comparar duas soluções: uma base de dados em memória persistente e uma em memória principal, neste caso, o SAP HANA. Concluiu-se que o SAP HANA apresenta melhorias, no que diz respeito ao tempo de execução de *queries*, nomeadamente quando nos referimos ao processamento analítico, isto é, de grandes e complexos volumes de informação. O objetivo do estudo foi demonstrar de que forma a migração para uma tecnologia deste género pode contribuir para a criação de valor num determinado negócio. O método de avaliação foi a medição dos tempos de execução de determinadas *queries*, que será o mesmo a utilizar nesta investigação.

3.2 Conclusões

De acordo com a revisão da literatura efetuada, concluiu-se que a utilização de uma base de dados em memória principal para efeitos de melhoria da gestão de inventário das organizações é algo ainda muito pouco explorado. A integração do negócio das empresas, com volumes gigantescos de informação, com dispositivos móveis é outro aspeto que maximiza a importância de sistemas em memória, capazes de responder em tempo real às consultas e, assim, flexibilizar e suportar a tomada de decisões, operacionais e executivas.

Nesta dissertação pretende-se realizar um caso de estudo na empresa em causa, combinando-o, posteriormente, com uma avaliação experimental. O estudo será efetuado com o sistema SAP HANA em particular e pretende mostrar de que forma poderá ser vantajoso utilizar uma solução desta natureza, para otimizar e melhorar a gestão de inventário da empresa e, assim, contribuir, decisivamente, para a criação de vantagem competitiva. A questão que se colocará é de que forma a base de dados HANA pode contribuir para esta vantagem competitiva, através da melhoria

Revisão da Literatura

da gestão do inventário da organização. No capítulo seguinte será abordada a metodologia de investigação, assim como o enquadramento do ambiente da experimentação.

Capítulo 4

Metodologia de Investigação

Os sistemas de informação são implementados numa organização com o objetivo de melhorar a sua eficácia e eficiência, contribuindo em muito para uma melhor gestão da informação e comunicação entre departamentos, refletindo-se na criação de valor para o negócio. As emergentes tecnologias de informação são um fator significativo na determinação das estratégias que guiam uma determinada empresa. Sistemas bem evoluídos e avançados, com o SAP HANA, permitem às organizações adotarem e optarem por outras estratégias, formas e estruturas, ou seja, alterarem a forma como conduzem o negócio. [HMPR04]

De seguida vai ser abordada a metodologia de investigação usada na realização da dissertação, com foco na ISRF, e será também formulada a respetiva pergunta de investigação.

4.1 Pergunta de Investigação

Em negócios críticos em tempo, adotar uma solução em memória parece ser o caminho para a criação de valor e vantagem competitiva. Contudo, uma abordagem desta natureza é um investimento substancial que as empresas devem estar dispostas a tomar, visando um sucesso vindouro no negócio, mas que deve ser utilizada adequadamente, de forma a retirar o máximo proveito desta solução inovadora. A crescente emergência dos dispositivos móveis, aliada à sua integração no negócio das organizações, vem reforçar e mostrar as vantagens duma solução deste tipo, pelo facto de conseguir responder em tempo real a um processamento gigantesco de informação. Assim, a pergunta de investigação formulada foi:

- De que forma pode, uma base de dados em memória, neste caso, o SAP HANA, contribuir para a criação de vantagem competitiva sustentável numa empresa, através da melhoria da gestão do inventário?

O foco está na otimização dos tempos de execução de consultas, de forma a flexibilizar e agilizar a tomada de decisões, indispensável na gestão de um negócio, criando uma vantagem sustentável em relação à concorrência.

4.2 Paradigma *Design Science - Information Systems Research Framework*

O paradigma *design-science* tem as suas raízes assentes na engenharia e nas ciências do artificial, sendo, fundamentalmente, um paradigma de resolução de problemas que procura criar inovações que definam ideias, práticas, capacidades técnicas e produtos através dos quais a análise, design, implementação e uso dos sistemas de informação possa ser eficientemente realizada.

Os artefactos criados devem ser avaliados de forma a resolverem problemas organizacionais [HMPR04] e, desta forma, serão executadas *queries* em ambos os sistemas, de forma a avaliar a utilidade de tecnologias em memória (SAP HANA) na criação de vantagem competitiva para a empresa. Contudo, criar artefactos úteis é um processo complexo, devido à necessidade de criar avanços nas áreas de domínio onde a teoria existente é, muitas das vezes, insuficiente. Estes estendem as barreiras da resolução de problemas humanos e das capacidades organizacionais, fornecendo ferramentas intelectuais e computacionais [HMPR04]. Neste caso, o artefacto a desenvolver corresponde a um módulo de *software*, uma aplicação *web* que permita gerir e acompanhar, em tempo real, o inventário de uma empresa de construção e, assim, avaliar o impacto das capacidades de um sistema em memória no negócio da organização.

As necessidades de uma organização podem ser definidas pelos objetivos, tarefas, problemas e oportunidades que são percebidos pelas pessoas dentro da organização. Estas necessidades são avaliadas no contexto das estratégias, estrutura e cultura organizacional, assim como, nos processos de negócio existentes.

Avaliações de artefactos estão, tipicamente, dependentes de métricas de desempenho, de forma a revelarem o quão bom é ou quão bem funciona [HMPR04]. Nesta situação, a métrica de desempenho será a medição dos tempos de execução de *queries* analíticas. O objetivo é estender a base de conhecimento atual com os resultados da investigação, de forma a tornar o conhecimento mais eficaz e permitir retirar ilações mais conclusivas.

A metodologia adotada nesta investigação foi a *IS Research Framework*, como pode ser observado na figura 4.1. Esta *framework* divide-se em Ambiente, Base de Conhecimento e Investigação. O ambiente no qual se baseia a investigação são organizações que possuem sistemas e processos não otimizados, como grandes volumes de dados que permanecem intocáveis e não têm utilidade. Estes tempos de resposta elevados levam, geralmente, a uma perda de vantagem competitiva no negócio da organização. Estas organizações fazem uso de tecnologias desatualizadas ou, sendo atualizadas, usam-nas de forma inapropriada, desperdiçando todas as potencialidades que tecnologias inovadoras, como a computação em memória, podem oferecer. Além disso, a integração com o novo paradigma da computação móvel, é algo muito importante a considerar e, organizações cujos sistemas não levem isso em conta, entram na esfera do ambiente inerente à investigação.

A base de conhecimento, necessária à realização da investigação, deve ser o mais completa possível, de forma a melhor suportar a investigação. Esta é formada por fundamentos, dos quais fazem parte teorias, *frameworks* e modelos, e metodologias, onde são referidas técnicas de análise de dados, assim como, formalismos e critérios de validação [HMPR04]. Neste caso, é fundamental

saber como funciona a tecnologia em memória e quais os seus principais conceitos e fundamentos, nomeadamente, a base de dados SAP HANA, quais as suas características diferenciadoras, de que forma pode criar vantagem competitiva para o negócio, assim como, otimizar sistemas já existentes que adotem uma abordagem desta natureza. A metodologia irá focar-se em analisar tempos de resposta de forma quantitativa, comparando a situação atual com a aplicação final desenvolvida. Os dados usados serão dados reais, fornecidos pelo cliente, uma empresa da indústria de construção. Mais detalhes sobre a informação usada podem ser consultados na secção 4.4.

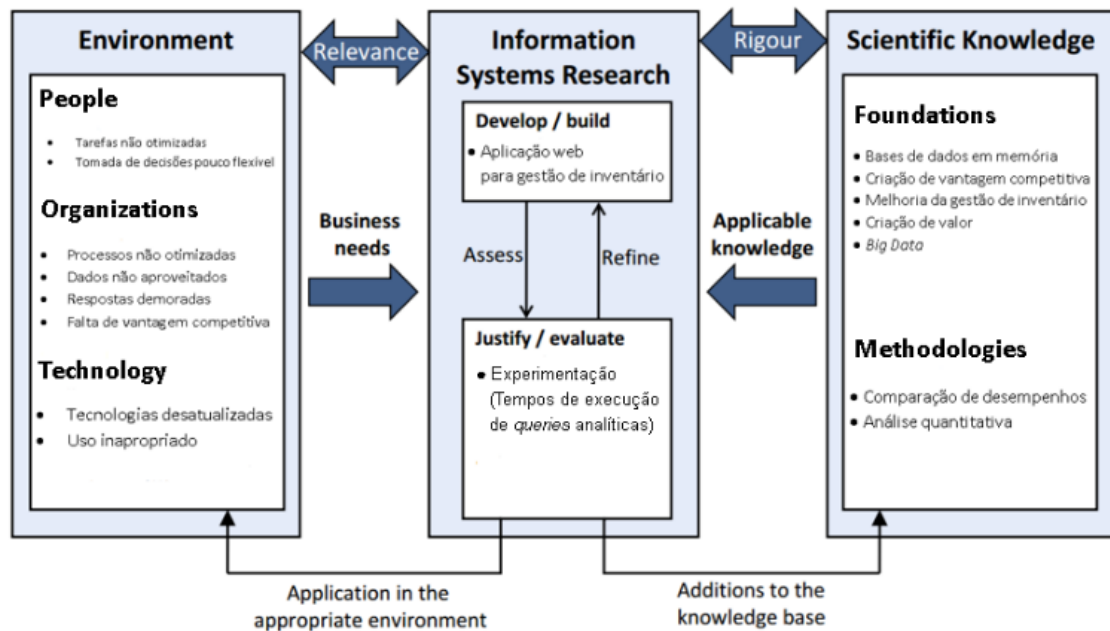


Figura 4.1: *Information Systems Research Framework*, adaptada de [HMPR04]

4.3 Enquadramento do ambiente de experimentação

Em ambientes industriais, a taxa de crescimento dos dados excede, geralmente, a capacidade dos seus sistemas de informação mais convencionais. Este crescimento de informação pode conduzir a tempos de espera muito altos, no processamento dos dados, o que se verifica no cliente em causa, refletindo-se numa tomada de decisões tardia, o que afeta o negócio das mais variadas formas, desde perda de oportunidade e consequente perda de dinheiro, culminando na perda de vantagem competitiva face à concorrência.

Apesar do cliente já possuir uma base de dados HANA, para onde foi migrada toda a informação, relativa à logística de materiais seriados e não seriados, a aplicação atual assenta num programa em SAP R/3 e as consultas aos seus movimentos de materiais revelam-se muito demoradas para resultados elevados (na ordem de horas ou nem mesmo exequíveis). Além disso, foram encontrados alguns problemas e incongruências na codificação do programa gerador destes relatórios que também contribuem para os tempos de resposta demorados do sistema atual. Consequentemente, a sua tomada de decisões é afetada e atrasada e problemas na gestão de inventário

começam a ser evidentes, por exemplo, a rastreabilidade dos produtos, que se revela importante na identificação das causas de produtos com defeito ou erros associados.

Pretende-se desenvolver uma aplicação gestora do inventário da empresa, utilizando a base de dados em memória SAP HANA e, desta forma, reduzir drasticamente os tempos de consulta aos movimentos de materiais, otimizando e flexibilizando todo o processo de tomada de decisões. Estas decisões constituem processos críticos no funcionamento e crescimento de um negócio e, portanto, a otimização destas consultas é um assunto de extrema importância para o sucesso futuro e criação de valor e vantagem competitiva para a organização.

4.3.1 Desafios de *Big Data*

Entramos na era da Informação, onde praticamente todas as empresas deparam-se com problemas associados a volumes gigantescos de dados (*Big Data*). Geralmente, este termo implica uma grande variedade de questões relevantes, como o tamanho, velocidade, complexidade e heterogeneidade dos dados [JKLS07]. Cada vez mais torna-se evidente a sua envolvimento nos mais variados campos, desde comércio e negócio à administração pública e investigação científica. O aumento do tamanho dos dados, associados à gestão de uma empresa, tem ultrapassado as capacidades de computação e para se ter uma ideia, o volume de dados empresariais por todo o mundo, na maioria das empresas, dobra a cada 1.2 anos. [PCZ14]

Para lidar com estes volumes gigantescos e complexos de dados, algumas técnicas foram surgindo, como armazéns de dados ou técnicas sofisticadas de *machine learning*, visando a exploração do conhecimento escondido por trás de toda esta informação [PCZ14]. De forma a tirarem partido de todos os benefícios que uma transição para uma estratégia de *Big Data* pode oferecer, as empresas têm de ser capazes de gerir esta mudança eficazmente. De acordo com [MB] são cinco as áreas fundamentais neste processo: Liderança, na importância de definir objetivos claros e fazer as perguntas corretas; Gestão de Talentos, devido à grande importância das capacidades dos engenheiros e cientistas da informação no uso de determinadas técnicas de análise; Tecnologia, uma vez que as ferramentas disponíveis para lidar com esta variedade, volume e velocidade de dados estão cada vez mais completas e poderosas; Tomada de decisões, visto uma organização eficiente colocar a informação e as decisões relevantes no mesmo local e Cultura da Empresa, onde a primeira pergunta que a empresa deverá colocar a si mesma é "O que é que sabemos?" e não "O que é que pensamos?".

O SAP HANA, plataforma de processamento em memória principal, permite um processamento bastante eficiente, idealmente em tempo real, de toda esta informação inerente à gestão e manutenção de um negócio empresarial.

4.3.2 Dados industriais (*shop floor data*) e criticidade de tempo

Os dados industriais, desde a manufatura à construção, geralmente referentes a quatro elementos comuns: máquinas, materiais, métodos e pessoas, tendem a crescer de forma abismal ao longo do tempo e, lidar e contornar esta situação tornou-se num dos problemas mais importantes na área

industrial [PC15]. Aplicações chave foram utilizadas, desde ERP, SCM e MES, e demonstraram-se muito úteis ao longo do tempo. Contudo, enfrentam desafios quando se trata de *Big Data*, assim como, determinados assuntos que permanecem não resolvidos, desde a flexibilidade dos requisitos como a capacidade de processar enormes volumes de dados, o que traz problemas no que concerne ao tamanho, velocidade, complexidade e heterogeneidade dos dados. [PC15]

A indústria de petróleo e gás, por exemplo, cria volumes gigantescos de dados com uma grande complexidade. É importante saber como lidar com tamanha informação, de forma a interpretar da melhor maneira os dados e, assim, conseguir suportar uma tomada de decisões em tempo real. Um sistema preparado para lidar com *Big Data* consegue criar uma imagem mais eficaz dos resultados prováveis (qualitativamente e quantitativamente) e esta habilidade permite, desde negociar um melhor preço, encontrar um comprador apropriado até um planeamento de logística otimizado. [SM06]

Respostas em tempo real e dados provenientes da *shop floor* compõem o esqueleto para a melhoria no processo de negócio e para a redução dos custos. Tempo é dinheiro e, portanto, quanto mais demorada uma determinada consulta, necessária à logística do negócio, maior a incapacidade de gerar produtividade e lucro na empresa, afetando a tomada de decisões, sejam elas feitas por um simples funcionário, por um administrador ou pelo CEO [Wau]. Em oposição, consultas eficientes permitem uma resposta rápida a ambientes adversos, tirando partido de oportunidades repentinas e, desta forma, cimentar uma posição vincada de vantagem competitiva.

Na última década, empresas de manufaturação automatizaram e computadorizaram vários processos, de forma a garantir alta produtividade, qualidade de produção e minimização dos custos associados. O objetivo é mostrar aos gestores aquilo que está a acontecer em tempo real no *shop floor*, fornecendo suporte na tomada de decisões com vista a maximizar a rentabilidade do negócio. [JKLS07]

4.3.3 Perspetiva de solução

De forma a processar esta quantidade enorme de *shop floor data* no menor tempo possível, uma abordagem em memória parece ser o caminho que oferece mais garantias para melhor alcançar este objetivo. Esta tecnologia facilita este cenário, devido à sua enorme capacidade de captura e armazenamento de grandes volumes de dados de uma forma oportuna. A análise e o processamento destas transações devem ser realizados em tempo real, ou perto disso, de forma a monitorizar os níveis de inventário e procura, assim como, permitir aos funcionários e administradores obter prontas respostas, criando vantagem competitiva para a organização. [vBDMU13]

Uma vez que o cliente já possui uma base de dados HANA, onde se encontra toda a informação necessária à produção dos relatórios de logística dos materiais (seriados e não seriados), pretende-se desenvolver uma aplicação que faça uso desta base de dados e maximize a otimização dos tempos de resposta das consultas inerentes ao funcionamento da organização.

Foi analisado e estudado todo o processo atual da empresa, incluindo o programa gerador dos relatórios, escrito em ABAP, linguagem utilizada no sistema SAP R/3, com o objetivo de reduzir drasticamente os tempos de consulta na nova aplicação, idealmente para a ordem dos segundos. A

aplicação *web* desenvolvida, para esta gestão e rastreabilidade dos materiais industriais, utilizou o *toolkit* SAPUI5 que permite a criação de aplicações *web* responsivas em *browsers* e dispositivos móveis, baseada em HTML5. Como forma de validação foram medidos os tempos de execução de *queries* analíticas, comparando o resultado obtido em ambos os sistemas e, assim, mostrando as vantagens da base dados HANA em relação a um sistema mais tradicional, o SAP R/3.

4.4 Experimentação

4.4.1 Dados

A metodologia foca-se em analisar, de uma forma quantitativa, os benefícios da base de dados HANA, ou seja, quantas vezes é melhor, relativamente à base de dados, de memória persistente, SAP R/3, de forma a dar resposta à pergunta de investigação. Toda a migração dos dados, de SAP R/3 para SAP HANA, usados neste estudo, foi executada pela empresa em causa, onde foram mantidos os nomes de todas as tabelas, bem como, as suas respetivas colunas.

O conjunto de testes compreenderá várias *queries* analíticas que serão executadas em ambas as bases de dados, onde se esperam obter tempos de execução significativamente inferiores, na base de dados HANA. Prevê-se que, com o aumento dos dados, ambos os tempos de execução aumentem, mas com a base de dados HANA a escalar, indubitavelmente melhor, do que a base de dados SAP R/3.

4.4.2 Origem e Tipo

Toda a informação usada nesta investigação foram dados reais, fornecidos pela empresa em causa, o que permite tirar conclusões e ilações mais concretas, uma vez que são dados efetivamente usados num negócio real podendo alargar o conceito a outras áreas dentro do setor empresarial. A empresa possui milhões de registos na sua base de dados SAP, relativos à logística do negócio, ambiente ideal para o uso de uma base de dados em memória principal, como a HANA.

A informação encontra-se distribuída por seis tabelas e diz respeito a movimentos de materiais efetuados numa determinada data. Cada registo possui informação como o documento a que pertence, a data do movimento, o material associado a esse movimento, o número de série caso seja um material seriado, o seu centro, o tipo de movimento, o lote, entre outra informação menos pertinente.

Conforme pode ser observado na figura 4.2¹, as tabelas principais são duas: MKPF e MSEG. A primeira diz respeito ao cabeçalho do documento relativo ao movimento do material e possui informação única, como a data de lançamento e o nome do utilizador associado ao registo do movimento. A segunda corresponde ao segmento (corpo) do documento associado a um determinado movimento. É nesta tabela onde se vai buscar grande parte da informação mostrada ao utilizador.

¹No diagrama UML, apenas estão representados os campos das tabelas mais importantes à compreensão da organização dos dados.

É também a tabela que possui mais atributos e a segunda maior cardinalidade de entradas e, por conseguinte, merecerá maior atenção que todas as restantes.

Cada movimento possui informação adicional como o nome descritivo do material ou do tipo de movimento, assim como, o número de série no caso de ser um material seriado. Esta informação encontra-se nas restantes quatro tabelas (ver figura 4.2²): a tabela MAKT, que contém o nome descritivo do material, assim como o respetivo idioma, a tabela T156T que contém o nome descritivo do tipo de movimento e o idioma e as tabelas SER03 e OBJK, ambas contendo informação necessária para alcançar o número de série associado a um determinado movimento.

Determinados atributos das tabelas, à exceção das chaves primárias, podem não conter nenhum valor para um movimento específico.



Figura 4.2: Diagrama UML representando a organização dos dados

4.4.3 Dimensão

A dimensão das estruturas de dados pode ser considerado o aspeto mais importante, uma vez que, uma base de dados em memória, tem como principal função processar e gerir enormes volumes de dados em tempo útil, reduzindo a sua complexidade e os tempos de execução associados a consultas a esses dados.

O número de registos das tabelas variam, sendo as tabelas MSEG e OBJK aquelas que possuem um maior número de entradas, conforme se pode constatar na tabela 4.1. Este tamanho é equiparado ao de uma média/grande empresa, com milhões de registos nas suas tabelas. Naturalmente, cada caso é um caso, e, conseqüentemente, as tabelas da base de dados podem assumir

²Breve explicação: MJAHR corresponde ao ano do documento, MBLNR ao número do documento, BUDAT à data de lançamento, USNAM ao nome do utilizador, MATNR ao número do material, ZEILE ao número do item no documento, LGORT ao depósito, WERKS ao centro, CHARG ao lote, BWART ao tipo de movimento, MAKTX ao nome do material, BTEXT ao nome do tipo de movimento, SERNR ao número de série e SPRAS ao idioma.

determinadas formas, tipos e complexidades, de acordo com as necessidades específicas do negócio da empresa.

Tabela 4.1: Número de registos das tabelas

Tabela	Número de registos
MKPF	3 475 358
MSEG	14 607 162
MAKT	107 528
T156T	27 168
SER03	5 169 663
OBJK	17 356 539

4.4.4 Validade

Para avaliar a aplicação em termos de desempenho, na execução das *queries*, foi muito importante o uso de dados reais, com um determinado tipo e tamanho característicos de ambientes empresariais reais. Deste modo, todos os negócios que se encaixem nestes moldes, podem, facilmente, tirar reflexões acerca da criação de vantagem competitiva para a empresa, ao adotar uma solução em memória, nuclear para uma transformação digital no dia-a-dia das empresas. Importante realçar que, se um destes dois fatores, referidos anteriormente, for alterado, os resultados esperados são diferentes, apesar da maior vantagem de desempenho do SAP HANA, em relação a uma solução de memória persistente, se demonstrar para gigantescos volumes de informação.

4.4.5 Método de Avaliação

Como forma de avaliar o desempenho da aplicação e das bases de dados SAP (HANA e R/3), irão ser medidos os tempos de execução de determinadas *queries* analíticas. O objetivo é mostrar as potencialidades de uma base de dados desta natureza e, de que forma, pode contribuir para a criação de vantagem competitiva sustentável e, consequente, criação de valor para o negócio de uma empresa. Assim, foi fulcral utilizar dados e *queries* reais, para uma análise mais completa e detalhada dos resultados, assim como, ser possível estender as conclusões a outras organizações, cujos dados, em tipo e em tamanho, se equiparem ao estudo realizado.

Durante a análise aos tempos de execução das *queries*, convém salientar o facto de que não existirá mais do que um pedido em simultâneo à base de dados, apesar de numa situação real, muito dificilmente, tal se verificar.

Grande parte dos *datasets* reais contêm valores distantes, chamados *outliers*, quer seja em excesso (valores distantes superiores) ou em escassez (valores distantes inferiores) comparados com os restantes valores da amostra. Estes valores podem ocorrer devido a fatores externos, como uma alta latência da conexão, inatividade da base de dados ou do servidor. Deste modo, importa detetar os valores distantes superiores, pois os inferiores ocorrerão em condições ótimas de teste. Vários métodos foram desenvolvidos para possibilitar uma análise mais rigorosa, em que foi escolhido o

método de *Tukey*, por ser mais resistente a valores extremos do que outros métodos investigados, como o método do desvio padrão ou o *z-score*. [Seo06, Dat]

O método de *Tukey*, proposto em 1977, é mais robusto a valores extremos devido ao uso de quartis. As suas regras são as seguintes:

- O IQR (*Inter Quartile Range*) é definido como a distância entre o quartil mais baixo (Q1) e o mais alto (Q3), isto é, a diferença entre Q3 e Q1;
- Se valor $< Q1 - 1.5 \cdot IQR$, então estamos perante um valor distante inferior;
- Se valor $> Q3 + 1.5 \cdot IQR$, estamos perante um valor distante superior e tal deve ser descartado;
- O primeiro quartil (Q1) corresponde a um valor superior a 25% dos dados, enquanto que o terceiro quartil (Q3) diz respeito a um valor superior a 75% dos dados. [Col, Seo06]

As *queries* analíticas serão executadas cinco vezes e o resultado será a média dos tempos de execução dessas cinco vezes, sem considerar valores distantes superiores. Se ocorrer algum valor distante, mas inferior, será considerado, pois significa que a conexão teve a menor latência possível e a inatividade da base dados teve a menor interferência possível. Desta forma, não se esperam variações significativas dos resultados.

Apesar da avaliação do artefacto desenvolvido, através da medição dos tempos de execução das *queries*, por si só não é suficiente para mostrar e provar todo o valor adjacente à plataforma SAP HANA. O objetivo é demonstrar de que forma uma base de dados em memória principal, como a HANA, pode contribuir para a criação de valor e vantagem competitiva sustentável para uma organização. O capítulo seguinte irá abordar os detalhes da implementação, assim como, a experiência conduzida, neste caso, as *queries* analíticas a serem executadas.

Capítulo 5

Implementação e Ensaio Experimental

O objetivo da implementação da aplicação, para gestão do inventário, é dar resposta, o mais eficaz e clara possível, à pergunta de investigação, ou seja, de que forma, pode o SAP HANA contribuir para a criação de vantagem competitiva e valor para a empresa, através da otimização e melhoria da sua gestão de inventário, neste caso, materiais seriados e não seriados. Com tempos de execução mais reduzidos às consultas, espera-se agilizar e flexibilizar todo o processo de tomada de decisões, associado aos dados industriais, criando oportunidade e vantagem competitiva sustentável para a organização. O foco passa por prover informação, em tempo real, aos recursos humanos, desde um simples funcionário ao CEO, informação essa necessária à gestão operacional e executiva do negócio.

Neste capítulo serão abordados detalhes da implementação, desde uma análise ao programa gerador dos relatórios em SAP R/3 até aos detalhes da aplicação desenvolvida em SAP HANA, assim como, as tecnologias e linguagens utilizadas. Será mostrada a experiência conduzida, com as respetivas *queries* analíticas a serem executadas em ambas as bases de dados, de forma a avaliar o seu desempenho e, de uma forma quantitativa, poder analisar o quanto uma se pode superiorizar a outra.

5.1 Implementação

A logística necessária ao funcionamento do negócio em causa assenta num programa desenvolvido em ABAP/4 no sistema SAP R/3. Esta aplicação geradora de relatórios exibe uma lista-gem que diz respeito a movimentos de materiais, com informações relativas a uma determinada operação, numa determinada data. No entanto, este programa revelou-se bastante demorado na execução de determinadas *queries* e, durante a sua análise, foram detetadas várias inconsistências ao nível da sua codificação. Perda de oportunidade de negócio e de foco na tarefa a realizar, uma tomada de decisões muito pouco flexível e uma consequente perda de dinheiro são alguns dos problemas vividos pela organização.

A ideia foi converter este programa, em SAP R/3, para SAP HANA e, desta forma, provar e mostrar a superioridade duma base de dados em memória perante uma base de dados centrada em disco, na execução de *queries* analíticas, num gigantesco volume de dados. A lógica do funcionamento do programa manteve-se praticamente inalterada, onde o principal objetivo foi a otimização dos tempos de execução das *queries*, idealmente para a ordem dos segundos. Entenda-se tempo de execução da *query* como o tempo desde que a *query* é enviada para o servidor da base de dados até ser recebida uma resposta. Como foi referido anteriormente, toda a migração das tabelas, de SAP R/3 para SAP HANA, foi realizada pela respetiva organização.

Ambos os sistemas foram conectados através de uma ligação remota, utilizando uma conexão via VPN. As bases de dados dos dois sistemas, HANA e R/3, encontram-se alojadas nas instalações da empresa, estando ambas as máquinas no mesmo servidor.

Apesar dos resultados apresentados ao utilizador serem os mesmos, as duas aplicações são diferentes, pois as linguagens de desenvolvimento diferem nos sistemas R/3 e HANA. Mais detalhes acerca da aplicação nos dois sistemas serão descritos nas próximas secções.

5.1.1 Aplicação em SAP R/3

Inicialmente, o foco foi estudar e analisar todo o programa que a empresa já detinha, de forma a facilitar a compreensão da origem dos dados (a que tabelas os dados iam ser procurados) e da sua seleção, isto é, a forma como os mesmos foram tratados, até serem mostrados ao utilizador. Esta fase foi particularmente útil aquando da replicação da aplicação para SAP HANA. O objetivo foi explorar o melhor possível as tabelas usadas, todos os atributos relevantes e identificar eventuais lacunas do programa, visando a sua otimização na conversão para SAP HANA.

Todo o processo foi custoso e tornou-se um pouco demorado, pelo facto da linguagem do sistema ser totalmente desconhecida até então e devido ao grande volume de código do programa estudado. Alguns atrasos com os acessos às respetivas máquinas da empresa em causa, contribuíram também para tal.

A aplicação consiste num ecrã de seleção composto por oito campos de seleção, nos quais o utilizador pode, ou não, inserir valores: número do material, centro, depósito, lote, tipo de movimento, data de lançamento, nome do utilizador e número de série. De acordo com os filtros colocados pelo utilizador a aplicação devolve uma lista de resultados com os movimentos de materiais que satisfaçam as condições inseridas. A aplicação é relativamente simples, mas de muita importância no negócio da organização e, portanto, a sua otimização tem um papel ímpar no sucesso e criação de vantagem competitiva da empresa.

Foi utilizado o SAP NetWeaver, que basicamente corresponde à interface gráfica para o utilizador em Windows, na versão 720. A linguagem de programação utilizada no programa analisado foi ABAP, onde o principal componente do sistema utilizado foi o *Debugger*, para uma melhor compreensão da lógica de funcionamento do programa, permitindo introduzir pontos de interrupção para análise do código.

5.1.2 Aplicação em SAP HANA

A aplicação desenvolvida usando a base de dados SAP HANA visa obter um desempenho muito superior à base de dados R/3, no que diz respeito à execução das *queries* analíticas necessárias ao funcionamento da organização. Uma vez que a migração já se encontrava efetuada pela empresa, todas as tabelas usadas já se encontravam povoadas e utilizadores com privilégios de acesso aos dados foram fornecidos pela organização.

Conforme pode ser visto na figura 5.1, cada campo de seleção possui duas caixas de texto. Se apenas a primeira for preenchida, esse campo terá de ter um valor igual ao inserido pelo utilizador; caso preencha apenas a segunda, o campo terá um valor até esse valor inserido no filtro; caso preencha as duas caixas de texto, esse campo terá de estar entre os valores especificados. Apenas a data de lançamento é um campo de preenchimento obrigatório pelo utilizador.

Figura 5.1: Visão da aplicação de logística de materiais em SAP HANA

Em termos de *design* da aplicação, uma única *query* é executada pela base de dados. Quando é feita uma pesquisa, um pedido AJAX é enviado ao servidor que comunica com a base de dados, executa a *query* e devolve os resultados à aplicação, para serem mostrados ao utilizador. A *query* é alterada dependendo dos filtros introduzidos pelo utilizador, de forma a devolver o conjunto de resultados correto e a diminuir ao máximo o tempo de execução das consultas, evitando *queries* extra para obter informação que não consta nas tabelas principais, MKPF e MSEG. Além do tempo de execução da *query* considera-se também o tempo de visualização dos resultados, desde que a aplicação recebe os dados da base de dados até serem efetivamente mostrados ao utilizador. O diagrama da figura 5.2 auxilia a compreensão da sequência de etapas, no funcionamento da aplicação.

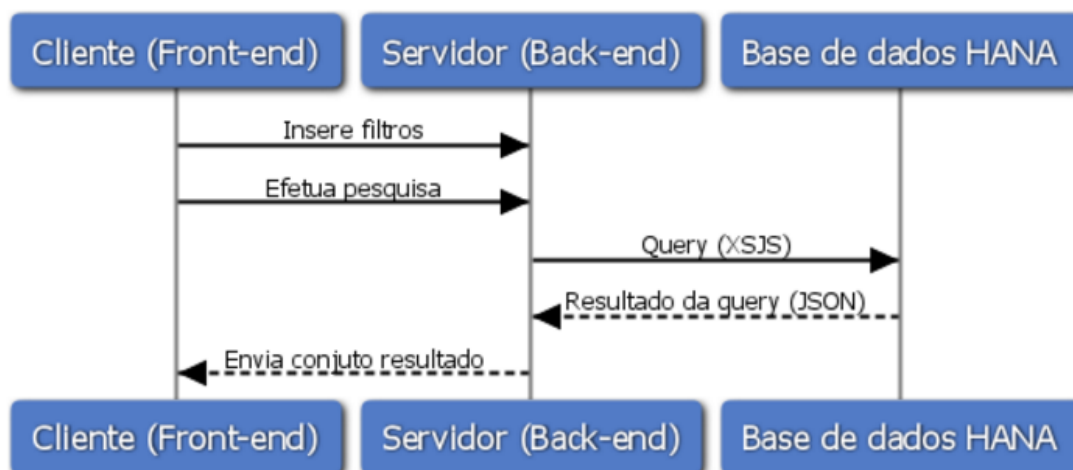


Figura 5.2: Diagrama de sequência que esquematiza o fluxo da aplicação

No desenvolvimento da aplicação foi utilizado o *toolkit* SAPUI5, que permite a criação de aplicações *web* com interfaces ricas para o utilizador, responsivas em *browsers* e dispositivos móveis, que se baseia em HTML5. Esta tecnologia assenta numa abordagem MVC (*Model-View-Controller*), onde os modelos representam a informação nas tabelas da base de dados, os controladores representam toda a lógica efetuada aos dados e as vistas constituem a interface que é mostrada ao utilizador final. Foi usado HTML5, Javascript (nos controladores) e XML (nas vistas). Como linguagem de interrogação à base de dados HANA foi utilizada SQL, executado pela linguagem do servidor HANA, XSJS (pequena variante de javascript). O ambiente integrado de desenvolvimento (IDE) usado foi o Eclipse, com os *plug-ins* para integração com uma base de dados HANA.

Esta base de dados, por possuir, maioritariamente, um formato de armazenamento orientado a coluna, consegue apresentar resultados bastante mais eficientes, sobretudo, em *queries* analíticas complexas. As *queries* transacionais não cabem no âmbito desta dissertação e, portanto, não foram consideradas.

5.2 Ensaio Experimental

Para avaliar o desempenho da aplicação desenvolvida, um conjunto de *queries* analíticas, de teste, serão executadas, visando a medição dos respetivos tempos de execução. O mesmo conjunto de testes será aplicado a ambas as bases de dados: R/3, onde assenta o atual programa da empresa e HANA, onde foi desenvolvida a nova aplicação otimizada. No sistema SAP R/3 os tempos de execução serão medidos recorrendo às funcionalidades do ERP, enquanto que, os tempos de execução em SAP HANA serão medidos através da ferramenta de desenvolvimento do *browser*, que entre outras coisas, fornece os tempos de execução das consultas, executadas pelo utilizador.

Os testes apenas serão analíticos, uma vez que, *queries* transacionais, como *INSERTS*, *UPDATES* e *DELETES* não cabem no âmbito desta dissertação, pois a aplicação não tem como objetivo

a alteração de dados. Deste modo, apenas serão executados *SELECTS* às bases de dados, que possuem tabelas com um gigantesco volume de registos. Serão pensadas *queries* que devolvam diferentes tamanhos de informação, para se poder fazer uma análise mais concisa ao comportamento da plataforma perante volumes de dados diferentes.

5.2.1 Queries de teste

As *queries* que constituem o conjunto de testes pretendem simular o uso real da aplicação para efeitos de logística de inventário. Assim, todas elas envolvem o mesmo número de tabelas (todas possuem informação a ser mostrada), onde são alterados os parâmetros da cláusula *WHERE*, conforme os filtros introduzidos pelo utilizador. O conjunto de testes terá de ser o mesmo em ambos os sistemas, para permitir uma comparação correta e justa dos tempos de execução das consultas.

Na tabela 5.1¹ podem ser consultadas todas as *queries* que compõem o conjunto de testes, com o respetivo tamanho do resultado (pequeno, médio, grande). Considerou-se um resultado pequeno até 20000 registos; médio, entre 20000 e 100000 resultados; grande, acima dos 100000 resultados.

Tabela 5.1: Conjunto de testes de *queries* analíticas

ID.	Query	Resultado
1	Movimentos associados ao centro 1400 no dia 13.01.2016	Pequeno
2	Na data 13.01.2016, a que material, depósito e centro pertence o número de série 20880214789	Pequeno
3	Informação relativa a todos os movimentos do dia 01.01.2015	Pequeno
4	No mês de Janeiro do ano 2016, quais os movimentos referentes ao intervalo de números de série de 1 a Z	Médio
5	Entre os meses Janeiro e Junho do ano 2015, movimentos registados pelo utilizador LOG_PERU	Médio
6	Entre os meses Janeiro e Junho do ano 2015, movimentos associados ao centro P211	Médio
7	No primeiro mês de 2015 quais os materiais (nome), movimentos (nome do tipo de movimento) e número de série associados	Médio
8	Movimentos de materiais nos meses Janeiro e Fevereiro do ano 2016	Grande
9	Movimentos de materiais do tipo 101, entre os meses Janeiro e Agosto do ano 2013	Grande
10	Movimentos de materiais do centro 1000 até ao 1300, entre os meses Junho e Dezembro do ano 2014	Grande

As três primeiras *queries* pretendem avaliar o desempenho da aplicação em consultas cujo conjunto resultado seja considerado pequeno (< 20000). As quatro seguintes avaliam a aplicação para um resultado considerado médio, sendo que as três últimas dizem respeito a um grande conjunto

¹Como a data de lançamento é um campo obrigatório, todas as *queries* de teste dizem respeito a uma determinada data ou intervalo de datas.

de resultados (> 100000). Para cada um destes grupos, as *queries* diferem nos filtros introduzidos pelo utilizador (com mais e menos filtros), de modo a avaliar se a inserção de filtros (desde um valor exato a uma gama de valores) afeta o comportamento da aplicação, para três diferentes tamanhos de resultados.

A *query* 1 pretende avaliar o desempenho da base de dados devolvendo apenas um único registo. Na *query* 2 a pesquisa é feita pelo número de série e a 3 apenas possui como filtro a data de lançamento (campo obrigatório). No que diz respeito às consultas com resultado médio, a *query* 4 faz uma pesquisa por intervalo de números de série num dado período. A 5 e a 6 alteram os filtros da pesquisa e a *query* 7 apenas possui a data como filtro. Nas três últimas *queries* alargou-se o período considerado, para obter mais resultados, onde as *queries* 9 e 10, ao contrário da 8, possuem mais filtros além do mandatário.

5.2.2 Servidores e erros esperados

As bases de dados usadas neste estudo, R/3 e HANA, encontram-se alojadas no mesmo servidor. Assim, uma análise a este servidor deve ser feita, descrevendo os seus principais detalhes e características, apesar de nem todos serem conhecidos.

O servidor encontra-se alojado nas instalações da empresa, utilizando uma solução HANA *on premise*, ao invés de uma solução na *cloud*, eliminando problemas como a dependência de conexão à *Internet* para acesso a dados, erros ou falhas que possam acontecer na *cloud* e uma maior limitação técnica no desenvolvimento de *software* complexo. O servidor onde executam ambas as bases de dados possui, aproximadamente, 500 GB de memória principal, possui dois discos de memória persistente, com capacidades de 1200 e 17000 GB e o processador possui 18 núcleos e 2 *threads* por núcleo com uma frequência de *clock* de 2300 MHz.

De forma a reduzir ao máximo a ocorrência de erros, todas as *queries* serão executadas cinco vezes e o tempo de execução será a média dessas cinco vezes. Estes erros podem originar variações inesperadas nos resultados e, portanto, todas as *queries* serão executadas sequencialmente. Isto aplica-se a ambos os sistemas, SAP HANA e SAP R/3. Fatores como a latência da conexão², inatividade da base de dados e elevado uso do servidor irão influenciar os resultados e poderão originar tempos de execução, indesejavelmente, superiores. Além do tempo de execução da *query*, é importante realçar o tempo de visualização dos dados (desde que o cliente recebe a informação até ser mostrada ao utilizador final).

Durante o análise e teste da aplicação em R/3, verificou-se a ocorrência de alguns erros na execução de determinadas *queries*, nomeadamente mensagens de *dump* de memória, ultrapassando os recursos computacionais disponíveis. Estas situações aconteceram, sobretudo, para grandes volumes de resultados, embora tenha sido detetado o mesmo erro para resultados considerados pequenos, o que revela as deficiências e inconsistências da aplicação. A má gestão de memória efetuada por este sistema também contribui decisivamente para tais falhas de desempenho.

²Durante a execução dos testes, a latência da conexão revelou valores muito reduzidos e portanto podem ser descartados.

Implementação e Ensaio Experimental

Espera-se que a base de dados HANA tenha tempos de execução inferiores à base de dados R/3 e, de uma forma quantitativa, conhecer o quão melhor e mais rápida é. No capítulo seguinte serão apresentados os resultados da experiência e reflexões acerca da possível criação de vantagem competitiva para o negócio, através da melhoria da gestão do inventário da organização.

Implementação e Ensaio Experimental

Capítulo 6

Discussão dos resultados

O ensaio experimental conduzido compreendeu um conjunto de dez *queries* analíticas que pretendem simular o uso real da aplicação de gestão de materiais do cliente, variando o número de resultados (pequeno, médio e grande) e os filtros aplicados. O objetivo é perceber de que forma ambas escalam com o aumento dos dados e com a inserção de filtros. Espera-se que a base de dados HANA apresente tempos de execução significativamente mais céleres, permitindo, de uma forma quantitativa, medir o quão melhor se comporta em relação a uma base de dados de memória persistente, neste caso, a SAP R/3. Neste capítulo são apresentados os respetivos resultados da experimentação, assim como uma discussão dos mesmos e são apresentados os principais fatores limitadores do estudo.

6.1 SAP R/3

No sistema SAP R/3 os tempos de execução das consultas foram medidos através da funcionalidade de medição de tempos de resposta do sistema. Foram detetados alguns erros aquando da medição dos tempos das consultas devido à má gestão e consequente falta de memória do servidor. Na tabela 6.1¹ é apresentada a média dos tempos de execução das *queries* para cada consulta.

Tabela 6.1: Média dos tempos de execução em SAP R/3

Query	Tempo Execução (ms)
1	102843,68
2	2659,4
3	98000
4	625000
5	108000
6	259000
7	540000

¹De notar que cada *query* em ambos os sistemas foi executada cinco vezes.

Discussão dos resultados

Query	Tempo Execução (ms)
8	Não devolve resultados
9	Não devolve resultados
10	Não devolve resultados

Conforme pode ser observado na tabela anterior, nenhuma das *queries* envolvidas no conjunto de testes teve uma resposta em tempo real e três dessas consultas não conseguiram mesmo obter quaisquer resultados. As *queries* 1, 3 e 5 revelaram tempos de execução muito similares, alcançando praticamente os dois minutos. Esperavam-se resultados um pouco inferiores para o primeiro grupo de *queries* (possivelmente em tempo real) onde os resultados são considerados pequenos (<20000), o que também se deve a falhas e inconsistências ao nível da codificação do programa. A *query* 2 foi a que apresentou melhor resultado, muito próximo de uma resposta em tempo real (ligeiramente acima dos dois segundos e meio). No grupo de consultas com resultado médio (>20000 e <100000) os tempos de resposta aumentaram consideravelmente e as *queries* 4, 6 e 7 refletem esse mesmo crescendo. Nas consultas com um resultado considerado grande (>100000) a base de dados comportou-se de forma muito negativa, não conseguindo sequer executar as *queries*, devolvendo um erro indicativo de recursos insuficientes. Para estes tempos de resposta elevados e para o *timeout*² do servidor nas últimas três consultas contribuíram as inconsistências e erros na codificação do programa em ABAP, assim como, a gestão pouco inteligente de memória efetuada por este sistema.

6.2 SAP HANA

Na aplicação *web* desenvolvida utilizando a base de dados em memória principal HANA, os tempos de execução foram medidos recorrendo às ferramentas de desenvolvimento do *browser*, neste caso, o *Google Chrome*. É importante perceber como o sistema escala com o aumento do volume da informação para, posteriormente, fazer uma análise comparativa dos tempos de execução em ambos os sistemas e tirar as respetivas conclusões. A tabela 6.2 apresenta a média dos tempos de execução no sistema de gestão de dados em memória abordado.

Tabela 6.2: Média dos tempos de execução em SAP HANA

Query	Tempo Execução (ms)
1	75,2
2	67,0
3	151,0
4	2240,0
5	1682,0
6	1374,0
7	6932,0

²*Timeout* da conexão ao servidor significa que este demora imenso tempo a responder a um determinado pedido de dados feito por outro dispositivo, identificando a ocorrência de um erro, apesar de não ser considerada uma mensagem de resposta.

Query	Tempo Execução (ms)
8	14850,0
9	8870,0
10	21970,0

De acordo com a tabela 6.2, todas as *queries* da bateria de testes executaram mais rápido do que no sistema SAP R/3 atualmente adotado pelo cliente. As três primeiras *queries* apresentaram os tempos de execução mais reduzidos, isto para um conjunto de resultados pequeno. Para um conjunto médio, à exceção da consulta 7, todas apresentaram respostas em tempo real. A *query* 7 apesar de ter um tempo médio de aproximadamente 7 segundos revelou-se bastante mais rápida do que no sistema anterior³. Regra geral os tempos de execução aumentaram com o aumento do conjunto de resultados, verificando-se os maiores tempos de execução para *queries* que envolvem os maiores conjuntos de resultados. Apesar das consultas 8, 9 e 10 caírem fora do âmbito de tempo real, as mesmas apresentam resultados bastante positivos (na ordem dos segundos), enquanto que em SAP R/3 as mesmas nem conseguiram ser executadas.

6.3 Validade e discussão

A base de dados HANA revelou um desempenho bastante superior à base de dados de memória persistente SAP R/3 no que diz respeito à execução de *queries* analíticas. O foco da dissertação foi o estudo deste tipo de *queries*, pelo que consultas transacionais (como *inserts*, *updates* ou *deletes*) não cabem no âmbito deste trabalho.

Utilizando uma base de dados centrada em disco (SAP R/3) nenhuma das consultas realizadas executou em tempo real (a mais próxima foi a *query* 2, ligeiramente acima dos dois segundos e meio). Por contraste, utilizando a base de dados HANA foi possível executar consultas até então impossíveis (*queries* 8, 9 e 10), revelando a extrema importância e valor duma base de dados desta natureza, uma vez que torna possível aceder a informação até então não disponibilizada ou disponível em tempos bastante elevados, levando ao *crash* do servidor por maximização do uso dos recursos computacionais.

Nenhuma das consultas em SAP HANA foi muito demorada, sendo a *query* 10 aquela com pior registo (cerca de 22 segundos). No sistema adotado, SAP R/3, o mesmo já não se pode afirmar: três *queries* não devolveram resultados e duas delas apresentaram tempos de execução significativamente elevados (*queries* 4 e 7).

Na tabela 6.3⁴ é apresentada a média dos tempos de execução de todo o conjunto de testes, onde se pode verificar, em termos globais, a maior rapidez e desempenho da base de dados HANA face ao seu sistema ancestral.

Em média, o desempenho da base de dados HANA foi cerca de 140 vezes mais rápido do que a base de dados de memória persistente, revelando uma otimização do programa gestor do

³A *query* 7 revelou-se cerca de 78 vezes mais rápida, em média, em HANA do que em R/3.

⁴Apenas foram consideradas as primeiras sete *queries*, uma vez que em SAP R/3 não foi possível executar as últimas três consultas.

Tabela 6.3: Média dos tempos de execução em ambos os sistemas

Base de dados	Tempo médio de execução (ms)
SAP R/3	247929,01
SAP HANA	1788,74

inventário da organização. Contudo, esta diferença poderia ser bem mais acentuada se os cálculos tivessem tido em conta as três últimas *queries*, onde os ganhos são imensos, devido à enorme capacidade deste sistema em lidar com grandes volumes de dados. Uma vez que não foi possível medir os tempos de execução das *queries* 8, 9 e 10 em R/3, estas também não foram consideradas no tempo médio em HANA, de forma à comparação ser o mais justa e fiável possível.

Outro aspeto relevante foi avaliar o desempenho das bases de dados para conjuntos de resultados de tamanho diferentes: pequeno, médio e grande, isto é, de que forma se comportam ambas as base de dados perante volumes de resultados de dimensões distintas. A tabela 6.4 apresenta a média dos tempos de execução para cada um destes grupos em ambos os sistemas.

Tabela 6.4: Média dos tempos de execução (ms) para os três tipos de resultados

	SAP R/3	SAP HANA
Pequeno	67834,36	97,73
Médio	383000	3057
Grande	Impossível	15230

De acordo com os resultados obtidos pode-se observar o crescendo dos tempos de execução consoante o crescimento do conjunto de resultados, mas com a base de dados HANA a escalar, indubitavelmente, melhor do que a base de dados R/3.

Para consultas que pertencem ao grupo pequeno de resultados (até 20000 registos), a base de dados HANA executou cerca de 700 vezes mais rápido, em tempo real, ao contrário da base de dados R/3, cuja média é cerca de 1 minuto. Para um conjunto médio de resultados, verificou-se a mesma situação, com a base de dados em memória principal a ter um desempenho 125 vezes superior à sua concorrente. O auge da superioridade da HANA em relação à R/3 verifica-se para grandes volumes de resultados (acima dos 100000), uma vez que em R/3 as consultas nem eram possíveis de ser realizadas, comparadas com uma média de 15 segundos em SAP HANA, revelando novas oportunidades de negócio de forma a adotar estratégias até então não conhecidas ou não possíveis. No gráfico da figura 6.1⁵ estão esquematizadas as variações nos tempos de execução consoante a dimensão do conjunto de resultados, de forma a facilitar a compreensão destes mesmos dados.

Os resultados introduzem um novo paradigma nas bases de dados e *design* de sistemas e a sua contribuição para a criação de valor para o negócio será visto no tópico seguinte.

⁵Por impossibilidade, para o sistema R/3 só foram considerados os dois primeiros grupos de resultados.



Figura 6.1: Tempo médio de execução das *queries*, para cada grupo de resultados

6.4 Criação de vantagem competitiva

Neste estudo, o propósito de uma base de dados em memória principal foi acelerar e otimizar o programa gestor de materiais do cliente em causa. Com isto pretende-se criar valor e consequente vantagem competitiva para a organização, flexibilizando a sua tomada de decisões e acelerando todo esse processo. Tal é possível devido aos tempos de resposta muito céleres (tempo real ou perto disso) que uma base de dados desta natureza oferece, mesmo para volumes gigantescos de informação, que permite às organizações adotar estratégias e criar vantagem competitiva de uma forma muito pouco explorada ou praticamente impossível no passado.

Na maioria dos casos, a base de dados HANA oferece tempos de resposta em tempo real, o que é um indicativo da confiabilidade e fiabilidade duma solução desta natureza na integração do negócio das organizações com o emergir da utilização de dispositivos móveis. Por contraste, a base de dados tradicional R/3 não apresentou resultados em tempo real para nenhuma das consultas e três delas não devolveram resultados de todo, excluindo a utilização deste tipo de sistema numa solução móvel para enormes volumes de informação.

A experimentação conduzida permitiu verificar e confirmar que soluções deste género são ideais para servir a integração do negócio com dispositivos móveis, devido à enorme capacidade de processar elevados volumes de dados em tempo real ou perto disso. Torna-se possível executar *queries* em SAP HANA que não foram possíveis executar em SAP R/3 e, portanto, há a possibi-

lidade de realizar determinadas consultas que até então eram impossíveis, como por exemplo, as *queries* 8, 9 e 10. Em média a base de dados HANA revelou-se 140 vezes mais rápida do que a base de dados R/3 na execução de *queries* analíticas.

Por forma a dar resposta à pergunta de investigação "De que forma pode, uma base de dados em memória, neste caso, o SAP HANA, contribuir para a criação de vantagem competitiva sustentável numa empresa, através da melhoria da gestão do inventário?", podemos afirmar que uma solução em memória revela-se inúmeras vezes melhor em relação a uma solução de memória persistente, permitindo executar determinadas consultas até então impossíveis de realizar (neste estudo, as *queries* 8, 9 e 10). Optando por uma base de dados em memória as organizações podem beneficiar de variadas formas:

- Maior rapidez na execução de consultas, permitindo executar algumas até então impossíveis, o que contribui para novas oportunidades e possibilidades de gerar lucro para o negócio, adotando estratégias até então inexecutáveis;
- Enorme flexibilidade e rapidez na tomada de decisões, fornecendo informação em tempo real ou perto disso aos seus administradores e consequente adoção de estratégias tendo em vista o sucesso da empresa;
- Possibilidade de integração com o ambiente móvel no negócio das organizações, devido aos tempos de resposta em tempo real oferecidos pela base de dados;
- Capacidade de processamento e utilização de volumes gigantescos de dados, muitas das vezes não aproveitados pelas organizações, mas que continuam a consumir recursos computacionais como capacidade de armazenamento.

Apesar das vantagens evidentes na criação de valor e vantagem competitiva para as organizações, a adoção de um sistema desta natureza é um investimento dispendioso e importa perceber se se justifica para um determinado negócio. Obviamente, negócios não críticos em tempo ou com uma quantidade de dados relativamente pequena não justificam tal mudança, assim como, negócios não dependentes de bases de dados ou sendo o seu uso muito reduzido. De seguida, são apresentadas as principais limitações do estudo efetuado.

6.5 Limitações do estudo

Apesar desta investigação poder não ser suficiente para dar uma resposta totalmente completa e esclarecedora do potencial e capacidades da base de dados HANA, pretende-se incrementar a base de conhecimento existente, no que diz respeito à gestão de dados em memória DRAM e à possibilidade de criação de valor para as empresas.

Variados fatores limitaram este estudo e impediram que o ensaio experimental pudesse envolver mais *queries* de teste ao desempenho do sistema. A total disponibilidade das máquinas do cliente, tanto do sistema SAP R/3 como do SAP HANA, nem sempre se verificou devido a manutenções necessárias da empresa. Um *upgrade* na máquina de desenvolvimento, durante a fase de

Discussão dos resultados

testes, manteve-a inoperacional durante algum tempo e contribuiu para uma limitação na fase de testes. Também o ataque informático vivido pelas organizações levaram-nas a tomar medidas de segurança e precaução, que passaram por desligar a conexão à Internet, de modo a evitar potenciais invasões aos dados ou contaminações de outras máquinas, o que impossibilitou, nesse período, o estabelecimento da ligação via VPN. Em [Mag] podem ser consultados mais detalhes deste tipo de ataque informático.

No que diz respeito à fase de testes, a má gestão de memória do sistema R/3 limitou este processo, originando *crashes* do servidor, sobretudo em consultas com grande número de resultados. Em SAP HANA tal não se verificou com a base de dados a conseguir executar todas as dez *queries* devolvendo resultados em tempo real ou tempos bastante aceitáveis.

Seria relevante estender o estudo a mais consultas de forma a simular mais casos reais e aprofundar a veracidade dos resultados, minimizando a ocorrência de erros. Devido a fatores externos mencionados anteriormente que contribuíram para a redução do tempo disponível na elaboração da fase de testes, o estudo ficou limitado a apenas dez *queries*. O alargamento do número de consultas de cada dimensão do conjunto de resultados (pequeno, médio e grande) poderá ser um ponto importante tendo em vista o melhoramento da experimentação conduzida.

Discussão dos resultados

Capítulo 7

Conclusões e Trabalho Futuro

Nos últimos anos, as bases de dados em memória principal têm ganho uma importância muito elevada (pelas características inerentes a esta tecnologia inovadora) e as organizações cada vez mais se mostram preocupadas e interessadas em adotar soluções desta natureza tendo em vista a maximização do lucro do seu negócio. A capacidade destes sistemas suportarem ambos os tipos de *workload*, desde processos de negócio analíticos e bem complexos (OLAP) até *workloads* transacionais e operacionais (OLTP) torna viável as agregações *on-the-fly* levando a uma simplificação dos modelos de dados e das aplicações empresariais, bem como da forma como os sistemas empresariais são concebidos. [SFL⁺12]

O desenvolvimento da aplicação *web* utilizando a base de dados HANA também comprovou esta simplicidade, onde foi utilizada maioritariamente javascript, uma linguagem largamente adotada, ao contrário da linguagem da aplicação R/3 (ABAP), que sendo totalmente nova compreendeu uma fase de formação e compreensão da mesma, para posterior análise do programa atualmente adotado pela empresa.

Os objetivos estabelecidos nesta dissertação foram alcançados, com a base de dados HANA a revelar-se cerca de 140 vezes mais rápida do que a base de dados de memória persistente SAP R/3. Uma vez que em R/3 não foi possível executar as consultas com um grande número de resultados, para este valor só foram consideradas consultas com um resultado pequeno e médio. Apesar dos volumes gigantescos de informação, a base de dados HANA revelou tempos de execução em tempo real ou muito próximos disso enquanto que a R/3 não conseguiu executar consultas com um grande número de resultados e nas restantes apresentou valores bastante superiores.

A bateria de testes permitiu verificar a criação de valor e vantagem competitiva para as organizações através de uma melhoria significativa nos tempos de execução de determinadas consultas. A oportunidade e possibilidade de execução de novas consultas permitirá adotar abordagens estratégicas diferentes, tanto do ponto de vista executivo como operacional. Decisões em tempo real podem ser tomadas e uma gestão muito mais eficiente dos recursos é agora possível. Estes testes

foram conduzidas de forma a complementar a base de conhecimento existente e provar a superioridade prevista de uma base de dados desta natureza face a uma solução em memória persistente.

Ficou claro que empresas que necessitem de tomar decisões regularmente no dia-a-dia ou que pretendam uma integração do seu negócio com um ambiente *mobile* devem adotar uma solução em memória principal, com vista à redução drástica dos tempos de execução. Além da facilidade e simplicidade da implementação de aplicações, novas consultas podem ser feitas à base de dados que até então não eram possíveis, como comprova o estudo realizado. A possibilidade de tomar decisões, sejam elas mais críticas ou menos, em qualquer hora e em qualquer lugar é possível usando uma tecnologia de gestão de dados em memória RAM.

7.1 Trabalho Futuro

A investigação acerca da computação em memória principal deve ainda ser aprofundada de forma a enriquecer e complementar a base de conhecimento existente e para eliminar eventuais dúvidas em relação à adoção desta tecnologia emergente.

A aplicação *web* desenvolvida é uma aplicação *responsive*, capaz de se adaptar a diferentes dimensões de ecrã (*desktop*, *smartphone* e *tablet*). Inicialmente a ideia seria convertê-la numa aplicação nativa *android* e *ios*, mas por falta de tempo foi impossível concretizar este aspeto. O foco da dissertação foi o estudo de *queries* analíticas, de forma a perceber e analisar o comportamento de uma base de dados em memória perante volumes tão enormes de informação. Estender o tipo de consultas efetuadas, de forma a enriquecer a base de conhecimento atual é outro ponto futuro a considerar.

Analisar e alargar a comparação a outras plataformas em memória principal e memória persistente seria interessante para reforçar o valor e aprofundar a veracidade das melhorias nos tempos de execução. Como referido no trabalho, nos dias de hoje, praticamente todas as organizações já se precaveram e possuem o seu próprio sistema gestor de dados em memória principal e seria relevante comparar uma solução SAP com uma não SAP, como Oracle ou Microsoft. Isto seria um estudo bastante interessante e importante que poderá ser conduzido no futuro, apesar do acesso a estes sistemas ser dispendioso e difícil.

Referências

- [Akr] Gaurav Akrani. Importance of Decision Making in Management, 2011. Acessado em 10-01-2017. URL: <http://kalyan-city.blogspot.my/2011/08/importance-of-decision-making-in.html>.
- [BDMB16] Gabriel Braga De Medeiros e Mota Borges. Study of SAP Hana in the in-memory context. 2016.
- [BTMF12] Joos-Hendrik Boese, Cafer Tosun, Christian Mathis e Franz Faerber. Data management with SAPs in-memory computing engine. *Proceedings of the 15th International Conference on Extending Database Technology - EDBT '12*, page 542, 2012. doi:10.1145/2247596.2247661.
- [Bun] Bunty Jain. SAP R/3 Overview, 2012. Acessado em 08-03-2017. URL: <https://pt.slideshare.net/seekbuntyjain/erp-sap-r3-overview-introduction/>.
- [Che] Xiaoming Chen. What is the difference between OLTP and OLAP?, 2014. Acessado em 02-11-2016. URL: <http://datawarehouse4u.info/OLTP-vs-OLAP.html>.
- [CJG15] Pravin Chandra, Anurag Jain e Manoj Kr Gupta. Query Optimization in Oracle. *Proceedings of the 41st International Conference on Very Large Data Bases*, pages 1770–1781, 2015.
- [Col] Colin Gorrie. Three ways to detect outliers, 2016. Acessado em 20-05-2017. URL: <http://colingorrie.github.io/outlier-detection.html>.
- [Cur] Curtis Franklin Jr. SAP HANA: Not Only In-memory Game In Town, 2015. Acessado em 10-01-2017. URL: <http://www.informationweek.com/big-data/software-platforms/sap-hana-not-the-only-in-memory-game-in-town/a/d-id/1320595>.
- [Dat] DataPig Technologies. Highlighting Outliers in your Data with the Tukey Method, 2014. Acessado em 20-05-2017. URL: <http://datapigtechnologies.com/blog/index.php/highlighting-outliers-in-your-data-with-the-tukey-method/>.
- [Den] Den Howlett. SAP's Q2 FY2015 - more color on the results, S/4 is a'coming, HANA revived, 2015. Acessado em 29-01-2017. URL: <http://diginomica.com/2015/07/21/saps-q2-fy2015-more-color-on-the-results-s4-is-acoming-hana-revived/>.

REFERÊNCIAS

- [DFI⁺13] Cristian Diaconu, Craig Freedman, Erik Ismert, Per-Ake Larson, Pravin Mittal, Ryan Stonecipher, Nitin Verma e Mike Zwilling. Hekaton: SQL Server's Memory-Optimized OLTP Engine. *Proceedings of the International Conference on Management of Data (SIGMOD)*, pages 1243–1254, 2013. doi:10.1145/2463676.2463710.
- [FCP⁺12] Franz Färber, Sang Kyun Cha, Jürgen Primsch, Christof Bornhövd, Stefan Sigg e Wolfgang Lehner. SAP HANA Database - Data Management for Modern Business Applications. *ACM Sigmod Record*, 40(4):45–51, 2012. doi:10.1145/2094114.2094126.
- [GVV13] Mohit Kumar Gupta, Vishal Verma e Megha Singh Verma. In-Memory Database Systems - A Paradigm Shift. *International Journal of Engineering Trends and Technology (IJETT)*, 6(6):333–336, 2013.
- [HMPR04] Alan R Hevner, Salvatore T March, Jinsoo Park e Sudha Ram. DESIGN SCIENCE IN INFORMATION SYSTEMS RESEARCH 1. *Design Science in IS Research MIS Quarterly*, 28(1):75–105, 2004.
- [JKLS07] J Jenkole, P Kralj, N Lavrac e A Sluga. A Data Mining Experiment on Manufacturing Shop Floor Data. In *Manufacturing Systems*, 2007.
- [KKZ99] Alfons Kemper, Donald Kossmann e Bernhard Zeller. Performance Tuning for SAP R/3. *IEEE Data Engineering Bulletin*, 22(2):32–39, 1999.
- [LL16] Per-Åke Larson e Justin Levandoski. Modern main-memory database systems. *Proceedings of the VLDB Endowment*, 9(13):1609–1610, 9 2016. doi:10.14778/3007263.3007321.
- [Mag] Sérgio Magno. Como evitar o ataque de ransomware, 2017. Acedido em 14-06-2017. URL: <http://exameinformatica.sapo.pt/noticias/software/2017-05-12-Como-evitar-o-ataque-de-ransomware>.
- [MB] Andrew McAfee e Erik Brynjolfsson. Spotlight on Big Data Big Data: The Management Revolution, 2012. Acedido em 15-03-2017. URL: <http://tarjomefa.com/wp-content/uploads/2017/04/6539-English-TarjomeFa-1.pdf>.
- [Nie] Jakob Nielsen. Website Response Times, 2010. Acedido em 10-12-2016. URL: <https://www.nngroup.com/articles/response-times-3-important-limits/>.
- [PC15] Jaehui Park e Su-young Chi. A Requirement for Traceability of Production Logs in Large-scale Shop Floor Data. In *Proceedings of the 2015 International Conference on Big Data Applications and Services - BigDAS '15*, pages 151–155, New York, New York, USA, 2015. ACM Press. doi:10.1145/2837060.2837084.
- [PCZ14] C. L. Philip Chen e Chun Yang Zhang. Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Information Sciences*, 275:314–347, 8 2014. doi:10.1016/j.ins.2014.01.015.
- [Pla] Hasso Plattner. In-Memory Data Management 2015 - The Inner Mechanics of In-Memory Databases, 2015. Acedido em 10-10-2016. URL: <https://open.hpi.de/courses/imdb2015>.

REFERÊNCIAS

- [Pla09] Hasso Plattner. A Common Database Approach for OLTP and OLAP Using an In-Memory Column Database. In *Proceedings of the 35th SIGMOD international conference on Management of data - SIGMOD '09*, pages 1–2, 2009. doi:10.1145/1559845.1559846.
- [Seo06] Songwon Seo. A review and comparison of methods for detecting outliers in univariate data sets. *Department of Biostatistics, Graduate School of Public Health*, pages 1–53, 2006.
- [SFGL13] Vishal Sikka, Franz Färber, Anil K Goel e Wolfgang Lehner. SAP HANA: The Evolution from a Modern Main-Memory Data Platform to an Enterprise Application Platform. *PVLDB*, 6(11):1184–1185, 2013.
- [SFL⁺12] Vishal Sikka, Franz Färber, Wolfgang Lehner, Sang Kyun Cha, Thomas Peh e Christof Bornhövd. Efficient transaction processing in SAP HANA database. *Proceedings of the 2012 international conference on Management of Data - SIGMOD '12*, 6(11):731–741, 2012. doi:10.1145/2213836.2213946.
- [SM06] Micheal Swartz e Shobhit Mathur. Feel the Power: Big Data, Big Opportunity - Is Your Commodity Value Chain Smart? *European Oil & Gas*, (June):22–23, 2006.
- [vBDMU13] Jan vom Brocke, Stefan Debortoli, Oliver Müller e Axel Uhl. In-Memory Database Business Value. Results from a Study on Retail Innovations. *360° - The Business Transformation Journal*, (7):16–26, 2013.
- [vBDRM13] Jan vom Brocke, Stefan Debortoli, Nadine Reuter e Oliver Müller. How In-Memory Technology Can Create Business Value: Lessons Learned from Hilti. *Manuscript submitted for publication*, 34(JANUARY), 2013.
- [Wau] Patrick Waurzyniak. Managing Factory Data. Manufacturing Software, 2012. Acedido em 01-01-2017. pages 71–82. URL: <https://manufacturingengineeringmedia.com>.

REFERÊNCIAS

Anexo A

Computação em memória - conceitos adicionais

A.1 Particionamento

Particionamento é o processo de dividir uma base de dados lógica em *datasets* independentes distintos. Partições são objetos da base de dados e podem ser geridos independentemente. A principal razão da aplicação do particionamento de dados é alcançar um paralelismo ao nível dos dados e, desta forma, permitir ganhos de desempenho. Uma vez que esta técnica é usada como uma etapa técnica para aumentar a velocidade das *queries*, esta deve ser transparente ao utilizador. Além dos ganhos em desempenho, o paralelismo melhora a disponibilidade e capacidade de gestão dos *datasets*. Não é de todo fácil encontrar o particionamento ótimo, pelo que existem várias técnicas para serem aplicadas em situações diferentes:

- Vertical - Divide os dados em grupos de atributos e é replicada a chave primária. Atributos geralmente acedidos juntos devem estar no mesmo *dataset*. Este tipo de particionamento é mais usado em bases de dados orientadas a colunas.
- Horizontal - Divide os dados em grupos de tuplos, de acordo com alguma condição especial. Existem quatro tipos de particionamento horizontal:
 1. *Range* - Divide de acordo com um chave de particionamento, que determina como os dados são divididos entre as partições.
 2. *Round Robin* - Esta forma de divisão revela-se rápida e confiável, dividindo uniformemente todos os tuplos pelas partições.
 3. *Hash-Based* - Esta técnica faz uso de uma função de *hash* para atribuir cada linha às partições. É indispensável um boa função de *hash*.
 4. *Semantic* - Separa os tuplos de acordo com determinadas especificações, como processos exclusivos que possuem poucos acessos a outras partições.

De forma a escolher a técnica correta de particionamento, é mandatório ter um conhecimento profundo da aplicação e da organização dos dados, assim como, avaliar a necessidade dos processos aceder aos dados de todas as partições. [Pla]

A.2 *Logging e Recovery*

Por forma a serem usadas em aplicações produtivas empresariais, as bases de dados necessitam de fornecer condições de durabilidade (letra D nas propriedades ACID). Para fornecer essa garantia, tolerância a falhas e alta disponibilidade têm de ser garantidas. O procedimento *standard* para permitir uma recuperação durável é o *logging*. Com a ajuda do *logging* e de protocolos de recuperação, as bases de dados podem voltar ao último estado consistente antes da falha ocorrida. Os dados são escritos em ficheiros de *log*, que são armazenados em memória persistente como discos rígidos ou SSD's.

Para lidar com o crescendo do volume dos dados e a intensificação dos *workloads*, sistemas empresariais modernos têm que escalar, usando múltiplos servidores dentro do ambiente do sistema empresarial. Com o crescente número de servidores e consequente crescimento do número de discos e CPU's, a probabilidade de falhas ao nível do *hardware* está a aumentar. Quando um servidor falha tem de ser reiniciado e restaurado ou outro servidor tem de tomar conta do *workload* do servidor onde ocorreu a falha. De qualquer forma, para restaurar o estado anterior do servidor antes da falha, os dados guardados em memória persistente têm de ser carregados novamente para memória principal. Este processo é chamado recuperação e assenta numa operação de *logging* do dicionário. É executado em duas tarefas subsequentes: ler os metadados e preparar as estruturas de dados e ler os dados de *logging* e recuperar a base de dados. [Pla]